



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Aprendizaje de modelos de interacción humano-robot con el robot NAO

Estudiante: Celtia Meizoso Lamas
Dirección: Francisco Javier Bellas Bouza
Berta María Guijarro Berdiñas

A Coruña, xuño de 2020.

A mi familia

Agradecimientos

En primer lugar quería agradecer a mis tutores, Francisco Javier Bellas Bouza y Berta María Guijarro Berdiñas, por toda su gran ayuda, apoyo y compromiso con este proyecto. También me gustaría agradecer al centro de mayores SUACASA, y en especial a la psicóloga del centro, Rocío Romero López, por proporcionarnos todo el conocimiento necesario para la realización de este trabajo. Por último, agradecer a mi familia y amigos por todo el apoyo que me han dado durante este proceso.

Resumen

La robótica de asistencia social (SAR) es un campo que actualmente está en auge debido a la necesidad de atender al creciente número de personas de edad avanzada y/o con necesidades especiales que requieren de cuidados. Uno de los objetivos principales, en estos casos, es tratar de que mantengan una vida independiente durante el mayor tiempo posible, evitando que sean ingresados en centros especializados. En base a esto se decidió desarrollar un sistema de monitorización de la vida diaria de una persona de edad avanzada con el robot humanoide NAO, al que se denominó SARDAM. La finalidad de este sistema es mantener al usuario activo el mayor tiempo posible y establecer hábitos saludables en la rutina diaria de la persona para reducir una aparición precoz del deterioro cognitivo y motriz. Para el diseño de este sistema y de la rutina diaria se consultó a especialistas en la atención a personas mayores, que proporcionaron toda la información necesaria para su construcción en base a una escala de prioridades. Integra sistemas de reconocimiento de personas, detección de objetos y reconocimiento del habla, además de las funcionalidades que incluye el robot. Un aspecto básico del sistema desarrollado ha sido la incorporación de un módulo de detección del estado de ánimo del usuario. Este módulo hace uso de la cámara y el micrófono del robot para detectar el estado de ánimo por la expresión facial y por las respuestas al interactuar por voz. En función de la emoción detectada, SARDAM propone al usuario unas actividades u otras, tratando de lograr el objetivo ya comentado de realizar las de mayor prioridad. SARDAM ha sido probado en un escenario, consiguiendo resultados lo suficientemente satisfactorios y opiniones positivas por parte del usuario.

Abstract

Social assistance robotics (SAR) is a field that is currently booming due to the need to cater to the growing number of elderly people and people who require intensive care. One of the main reasons of using these systems is to try to users independent for as long as possible, avoiding their admission to specialized centers. This project developed a system for monitoring the daily life of an elderly person with the NAO robot, called as SARDAM. The purpose of this system is to keep the user active and to establish healthy habits in their daily routine to reduce an early deterioration of cognitive and motor impairment. For the design of this system and the daily routine, specialists in the care of the elderly people were consulted and provided all the necessary information for its construction. SARDAM includes systems of people recognition, object detection and speech recognition, in addition to the functionalities included in the robot. A basic aspect of the system has been the incorporation of a

module which detects the user's emotional state. This module uses the robot's camera and microphone to detect the emotional state by facial expression and by the responses when interacting by voice. Depending on the emotion detected, SARDAM proposes to the user some of the activities, trying to achieve the objective of performing those with the highest priority. SARDAM has been tested in a scenario, achieving satisfactory results and positive opinions from the user.

Palabras clave:

- robótica de asistencia social
- interacción humano-robot
- inteligencia artificial
- robot NAO
- detección de emociones
- detección de objetos
- reconocimiento del habla

Keywords:

- socially assistive robotics
- human-robot interaction
- artificial intelligence
- NAO robot
- emotion detection
- object detection
- speech recognition

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Estructura de la memoria	3
2	Antecedentes	5
3	Fundamentos tecnológicos	9
3.1	Robot NAO	9
3.1.1	NAOqi	10
3.2	FaceNet	16
3.3	YOLO	16
3.4	FaceEmotion_ID	17
3.5	Google SpeechRecognition	17
4	Metodología	19
4.1	Planificación y gestión del proyecto	19
4.1.1	Planificación	19
4.1.2	Recursos	21
4.1.3	Gestión de riesgos	21
4.1.4	Estimación de costes	23
5	Desarrollo	25
5.1	Captura de requisitos	25
5.1.1	Primera captura de requisitos: Primera reunión con los investigadores	25
5.1.2	Segunda captura de requisitos: Primera entrevista con la experta	25
5.1.3	Tercera captura de requisitos: Segunda reunión con la experta	26
5.1.4	Requisitos funcionales	28

5.1.5	Requisitos no funcionales	28
5.2	Arquitectura de SARDAM	29
5.3	Detectores	31
5.4	Actuadores	32
5.5	Detección del estado de ánimo	32
5.6	Sistema de monitorización de tareas	40
5.6.1	Diseño de tareas	41
5.6.2	Diseño e implementación de SARDAM	45
6	Pruebas	59
6.1	Antes de empezar	59
6.2	Ejecución de las pruebas	64
6.2.1	Errores y problemas durante la ejecución	65
6.3	Opinión del usuario	66
7	Conclusiones	69
7.1	Líneas futuras	70
A	Documentación sobre las AVD	75
B	Descripción de las tareas	85
	Lista de acrónimos	91
	Glosario	93
	Bibliografía	95

Índice de figuras

2.1	Robot ENRICHME.	6
2.2	Robot GiraffPlus.	7
2.3	Características del robot Stevie.	7
3.1	Robot NAO V5 Evolution	11
3.2	Especificaciones del robot NAO	11
3.3	Comunicación entre los módulos a través del <i>broker</i>	14
3.4	Control cartesiano del movimiento del robot.	14
3.5	Posturas predefinidas del módulo ALRobotPosture.	15
4.1	Diagrama de Gantt de la planificación del proyecto.	20
5.1	Arquitectura del sistema SARDAM.	30
5.2	Zonas del NAO dónde se debe tocar para <i>DectTouch</i>	31
5.3	Ejemplo de resultados de la ejecución del sistema de detección de emociones por palabras.	37
5.4	Esquema gráfico del funcionamiento del sistema de detección del estado emocional.	39
5.5	Pseudocódigo de la plantilla de una tarea.	44
5.7	Ejemplo del fichero de configuración del horario.	45
5.6	Diagrama de flujo tarea genérica.	46
5.9	Ejecutar SARDAM.	47
5.8	Lista de ficheros del sistema SARDAM.	48
5.10	Diagrama de clases de SARDAM.	49
5.11	Rutina de ejercicios de movilidad física.	56
5.12	Ejemplo gráfico de ejecución de la actividad LibroDeVida con duración de 20 minutos.	58

6.1	Colocación del robot en la casa, con su campo de visión.	63
6.2	Prueba de la tarea Levantarse.	64
6.3	Ejercicio de movilidad física.	65
6.4	Fichero monitorizacion.log al finalizar el día.	65
6.5	Robot y usuaria interactuando entre ellos.	66

Índice de tablas

3.1	Articulaciones del robot NAO.	13
4.1	Riesgos capturados en el proyecto.	22
4.2	Evaluación de costes.	23
5.1	Resultados de la fusión de detección de emociones por rostro y tono de voz. .	35

Introducción

1.1 Motivación

Durante la última década han estado ocurriendo cambios demográficos a nivel mundial, especialmente en los países desarrollados, en los que destacan el aumento de la población de edad avanzada (una esperanza de vida más alta) y de otras enfermedades y minusvalías que necesitan de cuidados intensivos. Esto implica una creciente demanda de profesionales dedicados al cuidado de personas, que complica el poder proporcionar estos servicios, de manera apropiada, a quienes los necesitan. A raíz de ésto surge la robótica de asistencia social (*Socially Assistive Robotics*, SAR), con la que se pretende ofrecer este tipo de cuidados básicos pero a través de una máquina con capacidades de interacción con humanos.

El objetivo principal de la SAR es que la población que utilice estos servicios mantenga el mayor grado de autonomía posible, de manera que puedan seguir viviendo en sus hogares durante el mayor tiempo posible y así evitar ser internados en centros especializados. Destacan funcionalidades como la de ofrecer terapia física (ejercicio físico, rehabilitación) y asistencia en la vida diaria (recordatorios, sugerencias, etc.), así como animar y apoyar emocionalmente a las personas, entre otras [1]. SAR se clasifica en dos tipos: robots serviciales y robots de compañía o terapéuticos. En el primero, los robots tienen la finalidad de ayudar con las actividades de la vida diaria; en cambio, los robots de compañía se encargan de mejorar el estado psicológico y el bienestar de sus usuarios [2].

Como se comentó anteriormente, uno de los campos más destacados donde se está empezando a introducir esta tecnología es en el del cuidado de personas mayores. En él, se pretende que los robots ofrezcan servicios para el cuidado físico y cognitivo de la persona, monitorización y entrenamiento; es decir, tratar de aumentar su calidad de vida. Con esto principalmente se intenta evitar el aislamiento social y la soledad en la población anciana, ya que son unos de los mayores problemas que tiene este colectivo, llegando a ser causa de depresión e incluso de muerte, como está ocurriendo en países como Japón. A lo largo de los años se realizaron

varios estudios respecto a cómo dar una posible solución a lo dicho anteriormente y si era factible del uso de esta tecnología en este tipo de problemas [3]. En muchos de ellos, los resultados fueron positivos, especialmente a la hora de dar compañía al usuario y apoyarlo en su vida diaria y en su bienestar. Incluso, en centros de mayores o cuando las investigaciones se centraron en grupos que convivían juntos durante un intervalo de tiempo, se podía observar un incremento en las habilidades sociales de los residentes, como en el estudio de A. K. Ostrowski et al [4]. En él, las personas mayores se ayudaban entre ellas cuando alguien no sabía o no se acordaba de cómo interactuar con el robot Jibo [5]. También resultó útil el uso de notificaciones o alertas para situaciones de la vida diaria como tomarse la medicación, por ejemplo.

Cabe destacar que los robots diseñados para este tipo de funciones generan más empatía y afecto en el usuario respecto a otros dispositivos. Su aspecto visual suele ser el de un juguete, normalmente con rasgos humanos, como el NAO [6], o imitando al de los animales, como el PARO [7] o el AIBO [8]. Esto es importante para mayor confianza por parte de los usuarios de avanzada edad, ya que normalmente las nuevas tecnologías les suele causar rechazo.

La robótica de asistencia social para personas mayores es un sector que está en auge, y es de relevancia para los grupos de investigación LIDIA y GII de la UDC, promotores de este TFG. El objetivo es llevar a cabo una primera implementación de un sistema SAR basado en IA mediante el robot humanoide NAO, desarrollado en colaboración con los profesionales del sector asistencial, al que se le denomina como SARDAM (*Service Assistant Robot for Daily Activity Monitoring*). SARDAM permitirá, en el futuro, la monitorización autónoma de personas con bajo grado de dependencia, e incorporará capacidades de aprendizaje y adaptación al usuario. Finalmente, destacar la relevancia que este tipo de sistema puede tener en situaciones de crisis sanitaria como la vivida con el COVID-19, ya que reduce el contacto del personal sanitario con personas vulnerables.

1.2 Objetivos

La finalidad de este proyecto es crear un sistema de monitorización de la rutina diaria de una persona mayor utilizando un robot humanoide. No se pretende vigilar al usuario, sino sugerirle y motivarle para que cumpla con dicha rutina. El objetivo del sistema es mantener a la persona activa el mayor tiempo posible, y así ganar calidad de vida y evitar los deterioros cognitivos y motrices. Para ello, el sistema se apoya de un módulo de detección del estado de ánimo, que dependiendo de en qué estado se encuentre el sujeto, adaptará las actividades a sugerir. Todo el proceso de monitorización se basa en la interacción con el robot, que constituye el interfaz del sistema en la percepción y actuación.

Para lograr este objetivo global, se han propuesto los siguiente sub-objetivos:

1. Familiarizarse con el uso del robot humanoide NAO y de sus sensores, especialmente con la cámara, el audio y el movimiento.
2. Comprender y utilizar algoritmos de visión artificial para reconocer personas y objetos en el marco de los sistemas de interacción humano-robot.
3. Comprender y utilizar algoritmos de reconocimiento del habla en el marco de los sistemas de interacción humano-robot.
4. Revisar los principales algoritmos de reconocimiento del estado emocional basados en visión y audio.
5. Desarrollar un sistema de detección del estado de ánimo propio adaptado al robot.
6. Desarrollar un sistema que monitorice la vida diaria de una persona.
 - Capturar los requisitos del sistema en base a la consulta de expertos.
 - Diseñar e implementar un sistema que permita definir un plan diario de actividades.
 - Integrar los sistemas de percepción y actuación del robot como funcionalidades de dicho sistema, para así usarlas en la implementación de las actividades.
 - Permitir que se puedan añadir, modificar y eliminar actividades fácilmente.
 - Implementar dicho sistema de manera que ejecute las actividades dependiendo del estado de ánimo del usuario.
7. Realizar un experimento con el robot real NAO en un entorno de interacción con personas. Definir un escenario con el que se pueda comprobar como SARDAM, el sistema desarrollado, reacciona a ellos.

1.3 Estructura de la memoria

Los capítulos de esta memoria se estructuran como sigue:

1. **Introducción.** Se comenta qué aspectos fueron los que motivaron la realización de este TFG, y se hace una introducción a la robótica de asistencia social. También se exponen los objetivos de este proyecto.
2. **Antecedentes.** Se comentan los sistemas SAR más destacados relacionados con el objetivo de este proyecto.
3. **Fundamentos tecnológicos.** Se explican las principales tecnologías que se utilizaron para el desarrollo del sistema SARDAM.

4. **Metodología.** Se comenta cuál fue la metodología utilizada para el desarrollo de este TFG, y se muestra como se planificó y se gestionó el proyecto.
5. **Desarrollo.** Se explica cómo se obtuvieron los requisitos y cuáles son. Se explican cómo se desarrolló el sistema SARDAM y cómo se obtuvo cada uno de los elementos que lo componen.
6. **Pruebas.** Se comenta cómo se realizaron las pruebas, y cuáles fueron los resultados obtenidos y a qué conclusiones se llegaron.
7. **Conclusiones.** Se expone cuáles son las conclusiones sobre el sistema final desarrollado, y los objetivos cumplidos. Se comenta el trabajo futuro del proyecto.

Antecedentes

ACTUALMENTE se pueden encontrar varios proyectos centrados en la robótica asistencial dirigida a la población de la tercera edad. Como se ha mencionado en el Capítulo 1, existen dos tipos de robots asistenciales: los serviciales y los terapéuticos o de compañía. En este capítulo se revisarán los principales robots de servicio existentes en la actualidad, ya que esa es la finalidad del sistema propuesto en este TFG.

Uno de los primeros prototipos que salieron a la luz es el de *Nursebot project*, llamado Pearl [9]. Sus dos principales funcionalidades son la de recordar al usuario las actividades rutinarias que tiene que hacer, como comer, beber, tomar la medicación, usar el baño, etc., y la de guiarlo por su entorno, con especial hincapié en las actividades prioritarias (como tomarse la medicina). Además se intenta evitar que el robot no sea muy molesto o insistente con el usuario, y que tampoco el usuario sea demasiado dependiente de él. El software propuesto en este proyecto, Autominder, dispone de tres componentes principales: *Plan Manager* (PM), que almacena el plan diario de las actividades del usuario y es responsable de actualizar e identificar cualquier conflicto potencial en él; *Client Modeler* (CM), que utiliza la información sobre las actividades observables del cliente (usuario) para seguir la ejecución del plan; y *Personal Cognitive Orthotic* (PCO), que comprueba qué es lo que debería estar haciendo el cliente respecto a lo que está haciendo actualmente, y decide cuándo emitir recordatorios. El plan de las actividades diarias es diseñado por el cuidador, que también especifica cualquier preferencia sobre el tiempo o manera de ejecución. Se pueden añadir, modificar o eliminar las actividades de este plan diario.



Figura 2.1: Robot ENRICHME.

Otro prototipo más reciente es ENRICHME [10] (Figura 2.1), enfocado para la asistencia en casa. Este sistema permite realizar la monitorización de la salud, cuidados complementarios y el apoyo social para su usuario. Se utiliza una versión adaptada del robot TIAGo Iron, añadiéndole una *tablet*, una cámara térmica y el sistema RFID, entre otros. Este sistema es innovador respecto a otros trabajos anteriores debido a la mejora del multisensor de percepción humana gracias a la cámara térmica y el nuevo mapeado y localización de objetos basado en RFID. Para detectar personas hace uso de un detector láser para encontrar las piernas y de la cámara RGB-D para encontrar la parte superior del cuerpo, con la ayuda de la información que aporta la cámara térmica. También ofrece una aplicación en el dispositivo táctil incorporado en el torso, en el que el usuario puede escoger actividades como juegos cognitivos, consejos sobre salud, actividades físicas, búsqueda de objetos, datos del entorno, agenda, llamar a alguien, el tiempo y lectura de noticias.

GiraffPlus [11, 12] (Figura 2.2 ¹) es otro proyecto europeo enfocado en que sus usuarios puedan seguir manteniendo una vida independiente en su casa. El sistema consiste en una red de sensores ambientales y fisiológicos en casa que miden, por ejemplo, la presión sanguínea o el nivel de azúcar en sangre, o detectan cuando el usuario se cae, se sienta en una silla o a donde se dirige. Además, el robot es capaz de emitir alarmas o recordatorios al usuario o a sus cuidadores con respecto a dichas actividades. La plataforma robótica utilizada es el robot de telepresencia Giraff, que permite a los cuidadores y/u otros usuarios conectarse por videollamada con la persona mayor. También dispone de interfaces personalizadas para los usuarios primarios (la persona mayor) y secundarios (cuidadores) con las que se puede acceder y analizar la información obtenida por el sistema. La ventaja de este sistema es la capacidad de añadir y eliminar nuevos sensores perfectamente, y configurar automáticamente el sistema para diferentes servicios dados los sensores disponibles.

¹Imagen recuperada de [13]



Figura 2.2: Robot GiraffPlus.

En Galicia se está desarrollando el proyecto NETA V3A [14], donde participan las empresas Atendo, Balidea, GRADIANT, Ledisson y FrutoDs, con la finalidad de hacer un robot asistencial dirigido a la población gallega de la tercera edad. Su objetivo principal es ofrecer apoyo asistencial y social para su usuario para así prolongar la independencia y el estilo de vida en su entorno habitual. Se encargará de dar servicios como monitorización, fomentar hábitos saludables, apoyar tratamientos farmacológicos y médicos, facilitar el contacto social con el entorno del usuario (como videollamadas), etc. Se espera que este proyecto acabe a finales de 2020.

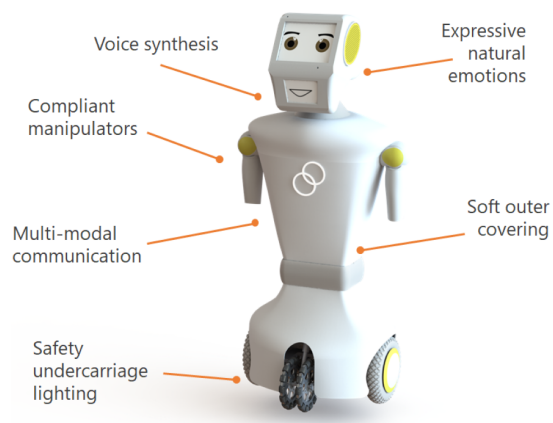


Figura 2.3: Características del robot Stevie.

También cabe mencionar a Stevie II (sucesor de Stevie) [15], lanzado en 2019, que sirve como un apoyo a los cuidadores y residentes de un centro. Es capaz de realizar tareas como recordar que deben tomarse la medicación, jugar con ellos, o incluso facilitarles acceso a nuevas

tecnologías. En la Figura 2.3 ² se pueden ver las características principales de este robot.

Con respecto al robot NAO, que será utilizado en este TFG, la empresa Zora Bots [17] ofrece desde hace unos años un software comercial llamado ZORA [18], que está dirigido más a centros que a hogares. Con él se pretende asistir al personal en el cuidado de niños y mayores, entre otras muchas funcionalidades. Su principal finalidad es la de acompañar y entretener, ya que a diferencia de los mencionados anteriormente, ZORA no ofrece monitorización u otro tipo de cuidados a sus usuarios; es más un robot de compañía que servicial. Dispone de una aplicación móvil con la que los cuidadores pueden seleccionar entre 50 actividades, juegos y bailes diferentes, hacer que el robot se mueva, ver y/o grabar con la cámara del robot, escribir el texto que se quiere que el robot diga a la velocidad y volumen que se desee, y programar el robot usando iconos de arrastrar y soltar. ZORA también puede dar clases de terapias físicas, leer en voz alta la programación de la televisión, el tiempo o las noticias.

²Imagen recuperada de [16].

Fundamentos tecnológicos

EN este capítulo se introducen las diferentes tecnologías más destacadas que formaron parte de este proyecto, tanto hardware como software. En concreto, se analiza en primer lugar el robot NAO que fue el empleado en el proyecto. Este robot humanoide proporciona una API llamada NAOqi que contiene todos los métodos necesarios para programar el robot y acceder a todas las funcionalidades que éste ofrece. Posteriormente se explica el modelo de reconocimiento facial FaceNet, que se utilizará en este proyecto para poder reconocer a personas. A continuación se comenta YOLO, que es un modelo para la detección de objetos eficiente para la detección a tiempo real. El siguiente modelo que se utilizó en el proyecto es FaceEmotion_ID, que es capaz de detectar siete emociones diferentes a través de una imagen dada de la cara de una persona. Por último se encuentra Google SpeechRecognition, que es una librería de Python que permite reconocer el habla de un audio con los diferentes sistemas que ofrece, en este caso el Google Speech Recognition.

3.1 Robot NAO

El robot que se utiliza en este proyecto es el **NAO** (figura 3.1) [6], desarrollado por la empresa francesa Aldebaran Robotics, subsidiaria de Softbank Robotics. La versión a usar es la V5 Evolution, que salió al mercado en el año 2014. La mayor ventaja de este sistema es su apariencia amigable con rasgos humanos que generarán empatía a la persona mayor, además de ya haber sido usado en muchas investigaciones sobre la temática de este TFG.

Este robot humanoide de 58 cm de alto es capaz de realizar movimientos y acciones complejas gracias a sus 25 grados de libertad. Su autonomía es de 60 minutos en uso activo, 90 minutos con un uso normal. Dispone de dos cámaras, dos altavoces, cuatro micrófonos, nueve sensores táctiles, dos sensores ultrasonidos, ocho sensores de presión, un acelerómetro y un giroscopio. Además incluye un sintetizador de voz, otorgándole una alta interactividad con el usuario. En la figura 3.2 se puede ver una imagen del robot con sus especificaciones indicadas.

La principal ventaja de NAO es que es muy fácil de programar, adaptable a cualquier tipo de usuario. Ofrece desde un software gráfico para programación en bloques, llamado Choreographe, a la posibilidad de programarlo con lenguajes como Python, C++, Java o JavaScript. Todo esto disponible para los sistemas operativos Windows, Linux y Mac.

Cuenta con un sistema operativo Linux basado en Gentoo, llamado NAOqi OS, que proporciona y ejecuta una gran cantidad de programas y bibliotecas, entre ellas, todas las requeridas por NAOqi, el elemento software que da vida al robot.

3.1.1 NAOqi

NAOqi es el software principal que se ejecuta en el robot y que permite la comunicación, programación e intercambio de información entre los diferentes módulos de movimiento, audio, vídeo que la componen. Para este proyecto se usa la API de NAOqi para Python 2.7.x.

Cuando el robot inicia se carga el fichero de preferencias llamado `autoload.ini`, que define qué librerías deben de cargarse. Cada librería contiene uno o más módulos que se utilizan en el *broker* (intermediario) para avisar a sus métodos. El *broker* (Figura 3.3) es el que se ocupa de la comunicación entre los distintos módulos. Proporciona servicios directorio (permitiendo encontrar los módulos y métodos), y acceso a la red (dando la posibilidad a los métodos de los módulos adjuntos de poder ser llamados fuera del proceso actual).

Para acceder a los módulos que ofrece este software, se hace uso de un proxy. Existe un módulo llamado `ALProxy` que permite crear el proxy de un módulo y así obtener un objeto que contiene todos sus métodos. Los parámetros requeridos son el nombre del módulo, la IP del robot y el puerto a un *broker*. Por ejemplo, para obtener el proxy de módulo `ALTextToSpeech` haríamos lo siguiente:

```
1 from naoqi import ALProxy
2
3 tts_proxy = ALProxy("ALTextToSpeech", IP, PUERTO)
4
5 # Con este objeto ya se pueden acceder a los métodos de
6 # ALTextToSpeech. Por ejemplo, para que el robot diga algo:
7 tts_proxy.say("Hola. Probando a hablar.")
```

Respecto a los módulos, pueden ser tanto remotos como locales. Si es remoto, es compilado como un fichero ejecutable, y puede ser ejecutado fuera del robot; son más fáciles de usar, pero son menos eficientes en términos de velocidad y uso de memoria. Si es local, es compilado como una librería y solo puede ser usado en el robot. Su principal ventaja es que son más eficientes que los remotos.

A continuación se describen los módulos utilizados (a parte de `ALProxy`) en este proyecto:

- NAOqi Core. En él se encuentran todos los módulos de propósito general, como el que



Figura 3.1: Robot NAO V5 Evolution

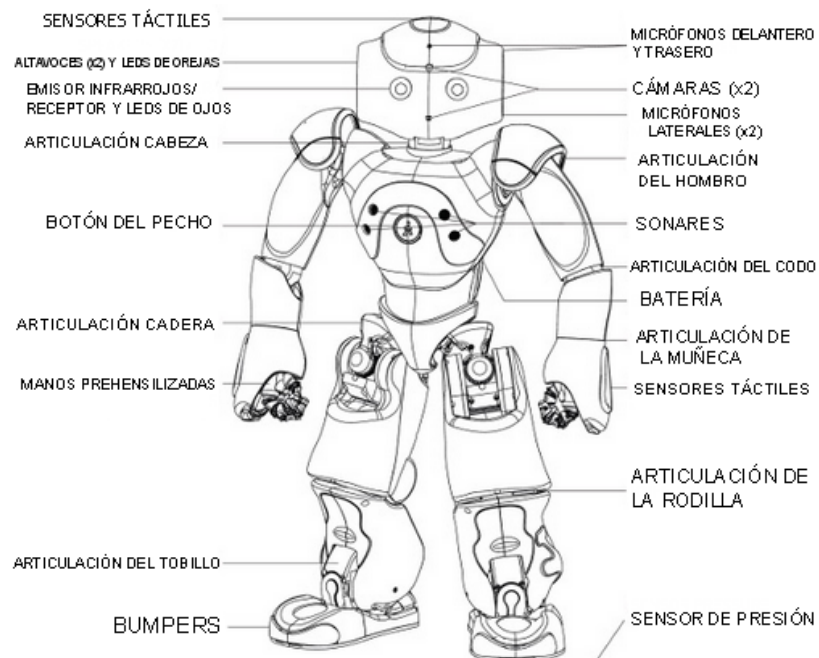


Figura 3.2: Especificaciones del robot NAO

maneja el sistema del robot (ALSystem), el relacionado con la vida autónoma del robot (ALAutonomousLife), que mantiene el robot activo y sensible a lo que sucede a su alrededor, o el que gestiona y configura conexiones a la red (ALConnectionManager). Los dos módulos que se usan de esta API son:

- **ALMemory**. Es la memoria del robot. En ella se guarda toda la información relacionada con la configuración hardware del robot (Actuadores y Sensores). Ofrece una serie de funciones para el almacenamiento y recuperación de dicha información.
- **ALModule**. Clase de la que heredan los módulos propios (del usuario) para anunciar qué métodos se desea que estén disponibles para los usuarios. Tanto este módulo como ALProxy son necesarios a la hora de crear un módulo nuevo.
- NAOqi Motion. Contiene todos los módulos relacionados con la movilidad del robot.
 - **ALMotion**. En él se encuentran todos los métodos que se encargan del movimiento del robot, clasificados en cuatro grupos: los que controlan la rigidez de las juntas, los relacionados con la posición de las articulaciones, los que permiten que el robot camine (control de la locomoción), y los que controlan el robot en el espacio cartesiano. De los cuatro el único que no se ha usado es el que controla la locomoción del robot, ya que el robot siempre se encontrará en una posición estática.

En el primer grupo que controla la rigidez del robot se encuentran los métodos *wakeUp()*, que enciende el motor (y la rigidez) y coloca el robot en la posición inicial, y *rest()*, que apaga el motor (y la rigidez) y se coloca en una posición relajada y segura. Estos métodos son muy importantes y útiles a la hora de querer mover el robot. La rigidez de una articulación es la fuerza de torsión necesaria para mover las articulaciones, y está relacionada con los motores. Es decir, si la rigidez de la articulación está a 0.0 (el valor mínimo), entonces dicha articulación no se puede mover; en cambio, si es 1.0 (valor máximo), entonces la articulación puede utilizar toda la potencia del motor para alcanzar la posición deseada.

El control de las articulaciones, tanto individualmente como un grupo de ellas, se puede hacer especificando los ángulos (*setAngles()*), en radianes, que se quieren rotar de la articulación, o por interpolación de ángulos (*angleInterpolation()*). Este último es útil para cuando se quiere hacer una trayectoria determinada con más de una articulación, ya que hace el código mucho más simple.

El control cartesiano también se utilizó en el proyecto. Consiste en mover la articulación para alcanzar el objetivo marcado en el espacio cartesiano. En este grupo

los métodos controlan los efectores del robot usando uno de los dos tipos de solucionador de cinemática inversa: el clásico que sólo utiliza las articulaciones de la cadena de efectores para alcanzar el objetivo, y uno generalizado que usa todas las articulaciones del robot para lograr llegar a la posición. Un efector es un punto 3D predefinido en el robot. En caso de utilizar alguna de las funciones de este grupo, hay que tener en cuenta en cuál de los tres espacios se basa la trayectoria definida: FRAME_TORSO, FRAME_ROBOT y FRAME_WORLD. En la Figura 3.4¹ se puede ver cuáles son los efectores del robot (Figura 3.4a) y los tres tipos de espacios que se mencionaron anteriormente (Figura 3.4b).

En la Tabla 3.1 se listan todas las articulaciones que componen el robot. Las articulaciones giran en torno al eje x (*roll*), eje y (*pitch*) y eje z (*yaw*).

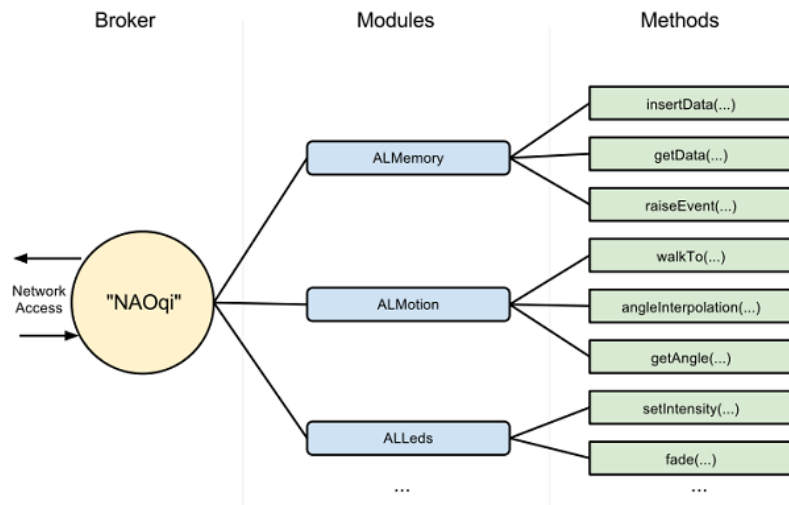
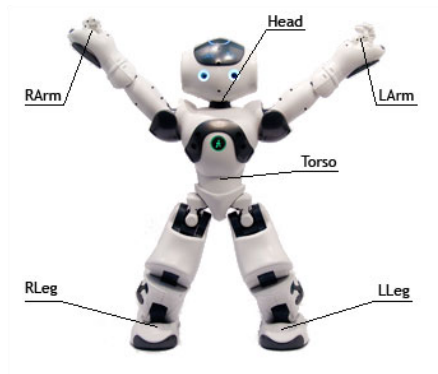
	Nombre articulación	Grados de movimiento (°)
Articulaciones de la cabeza	HeadPitch	-38,5 a 29,5
	HeadYaw	-119,5 a 119,5
Articulaciones del brazo izquierdo	LShoulderRoll	-18 a 76
	LElbowRoll	-88,5 a -2
	LElbowYaw	-119,5 a 119,5
	LWristYaw	-104,5 a 104,5
	LShoulderPitch	-119,5 a 119,5
Articulaciones del brazo derecho	RShoulderRoll	-76 a 18
	RElbowRoll	2 a 88,5
	RElbowYaw	-119,5 a 119,5
	RShoulderPitch	-119,5 a 119,5
Articulaciones de la cadera	LHipYawPitch	-65,20 a 42,44
	RHipYawPitch	-65,62 a 42,44
Articulaciones de la pierna izquierda	LHipPitch	-88 a 27,73
	LHipRoll	-21,74 a 45,29
	LAnkleRoll	-27,79 a 44,06
	LKneePitch	-5,29 a 121,04
	LAnklePitch	-68,15 a 52,86
Articulaciones de la pierna derecha	RHipPitch	-88 a 27,73
	RHipRoll	-45,29 a 21,74
	RAnkleRoll	-44,06 a 22,80
	RKneePitch	-5,90 a 121,47
	RAnklePitch	-67,97 a 53,40

Tabla 3.1: Articulaciones del robot NAO.

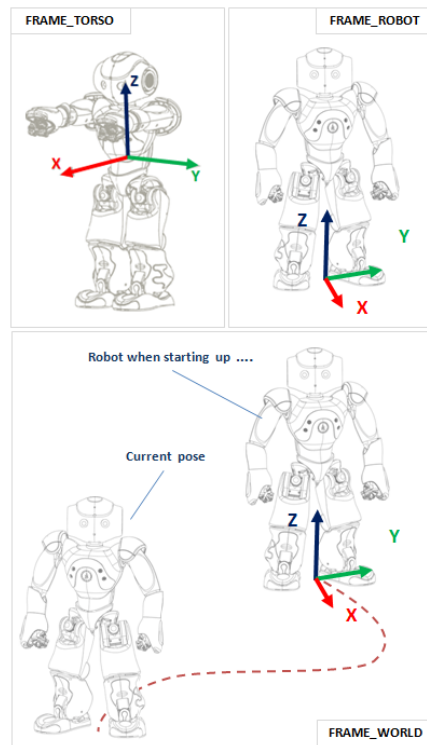
- **ALRobotPosture.** Permite que el robot haga cualquiera de las posturas predefinidas (Figura 3.5², obtenida de la documentación del NAO).

¹Imágenes obtenidas de la documentación de NAO

²Imagen recuperada de la documentación de NAO

Figura 3.3: Comunicación entre los módulos a través del *broker*

(a) Efectores del robot.



(b) Espacios en los que se puede mover el robot.

Figura 3.4: Control cartesiano del movimiento del robot.

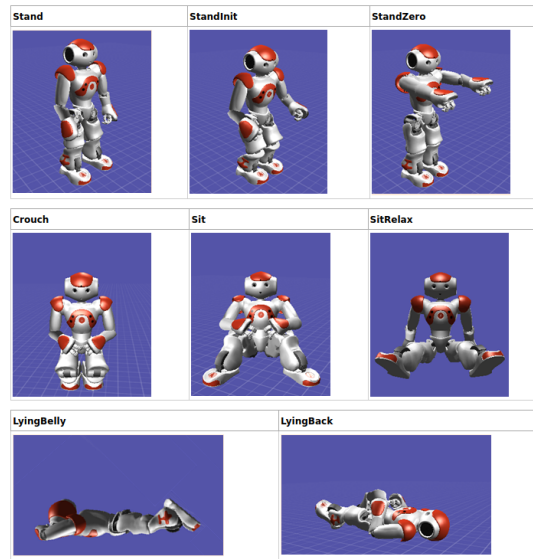


Figura 3.5: Posturas predefinidas del módulo ALRobotPosture.

- NAOqi Audio. Es donde están los módulos que se encargan de gestionar y detectar el sonido.
 - **ALSpeechRecognition**. Permite que el robot pueda reconocer palabras o frases predefinidas en el lenguaje seleccionado.
 - **ALTextToSpeech**. Permite que el robot hable. Envía comandos a un sistema *text-to-speech*, y autoriza la personalización de la voz. El resultado de la síntesis es enviado a los altavoces del robot. Se puede seleccionar el idioma en el que se quiere que el robot se comunique.
 - **ALAudioPlayer**. Reproduce audios en el robot.
 - **ALAudioRecorder**. Graba audios en el robot.
- NAOqi Vision. En él están todos los módulos que se encargan de todo lo relacionado con las cámaras del robot, además de detección y reconocimiento.
 - **ALVideoDevice**. Permite obtener imágenes de las cámaras del robot y guardarlas en disco.
 - **ALVideoRecorder**. Graba vídeos en un intervalo de tiempo.
- NAOqi People Perception. Sus módulos se utilizan para analizar a las personas que hay alrededor del robot.
 - **ALBasicAwareness**. Permite que el robot reaccione a estímulos que se encuentren a su alrededor. En el caso de este proyecto, se utilizará para detectar si hay

alguien a la vista del robot o no.

- NAOqi Sensors. Están todos los módulos relacionados con los sensores del robot.
 - **ALTouch**. Este módulo genera el evento “TouchChanged” cada vez que se toca el robot, devolviendo una lista con las partes del robot que fueron tocadas.

3.2 FaceNet

Una de las funcionalidades de las que necesitamos dotar al robot es el reconocimiento de personas. Para ello utilizamos el sistema de reconocimiento facial desarrollado por investigadores de Google en 2015, **FaceNet** [19]. Para la integración de esta tecnología en este TFG nos basamos en el tutorial *How to Develop a Face Recognition System Using FaceNet in Keras* [20]. Es un modelo *one-shot*, que aprende directamente un mapeo de imágenes de caras a un espacio euclídeo compacto donde las distancias corresponden directamente a una medida de similitud de rostros. Con una imagen dada, el sistema extrae las características de alta calidad de la cara detectada y predice una representación de ellas en un vector de 128 elementos, denominado *face embedding* (incrustación de la cara). De esta manera, al calcular la distancia entre los vectores de dos imágenes, si las caras son de la misma persona tendrán distancias pequeñas entre ellas mientras las que son de distintas personas serán más grandes. Con este método, el reconocimiento de caras es fácil de hacer; con métodos de Aprendizaje Automático como k-vecinos más cercanos o máquinas de vectores de soporte (*Support Vector Machines*, SVM) se puede reconocer a la persona detectada. Este sistema destaca respecto a los demás por su eficiencia, obteniendo una precisión del 99,63% en el dataset “*Labeled Faces in the Wild*” y un 95,12% en “*Youtube Faces DB*”, un porcentaje mayor al de otras aproximaciones.

3.3 YOLO

Otra de las habilidades que necesitamos añadir a NAO es la capacidad de la detección de objetos. Para ello se utiliza YOLOv3 [21]. Implementado oficialmente por Darknet, es un algoritmo que puede detectar objetos usando una única red convolucional, haciendo esta tarea de manera muy rápida. Ésta es una versión mejorada de sus predecesoras YOLO y YOLOv2. Su uso es óptimo en detección a tiempo real.

Como se ha dicho anteriormente, este modelo destaca por ser más rápido y exacto que otras implementaciones. Divide la imagen en regiones y predice las cajas delimitadoras y probabilidades por cada objeto detectado en cada región. Como dice su nombre (YOLO - “*You Only Look Once*”), las predicciones son hechas desde una visión global de la imagen, en tiempo test, siendo una de sus mayores ventajas. Además, gracias a su única red de evaluación llega

a ser mucho más rápido que otros modelos como el R-CNN (1000 veces más rápido) o el Fast R-CNN (100 veces más rápido).

Para incluirlo en este TFG se siguió el tutorial *How to Perform Object Detection With YOLOv3 in Keras* [22]. En el modelo que se utiliza en dicho tutorial y, por ende, en este proyecto, es capaz de detectar los siguientes objetos: “persona”, “bicicleta”, “coche”, “motocicleta”, “avion”, “bus”, “tren”, “camion”, “barco”, “semaforo”, “boca de incendio”, “señal de stop”, “parkimetro”, “banco”, “pajaro”, “gato”, “perro”, “caballo”, “oveja”, “vaca”, “elefante”, “oso”, “cebra”, “jirafa”, “mochila”, “paraguas”, “bolso”, “corbata”, “maleta”, “frisbee”, “skis”, “snowboard”, “pelota”, “cometa”, “bate beisbol”, “guante beisbol”, “skateboard”, “tabla de surf”, “raqueta de tenis”, “botella”, “copa”, “vaso”, “tenedor”, “cuchillo”, “cuchara”, “plato”, “platano”, “manzana”, “sandwich”, “naranja”, “brocoli”, “zanahoria”, “perrito caliente”, “pizza”, “donut”, “pastel”, “silla”, “sofa”, “plantas en maceta”, “cama”, “mesa”, “baño”, “monitor de televisión”, “portatil”, “raton del ordenador”, “mando”, “teclado”, “telefono movil”, “microondas”, “horno”, “tostadora”, “lavabo”, “refrigerador”, “libro”, “reloj”, “florero”, “tijeras”, “osito de peluche”, “secador”, “cepillo de dientes”.

3.4 FaceEmotion_ID

Otra de las funcionalidades necesarias es el reconocimiento de emociones en el rostro. Para ello se usa una versión adaptada de **FaceEmotion_ID** [23]. Este sistema se basa en el modelo CNN propuesto por Octavio Arragia et al., llamado mini-Xception [24]. Destaca por su eficiencia para programas en tiempo real, y por el uso de módulo residuales y convoluciones separables en profundidad, que reducen la computación con respecto a las convoluciones convencionales al reducir el número de parámetros.

Para este TFG se decidió hacer uso de un modelo preentrenado con la base de datos FER2013 que se encontraba en el repositorio de Github de FaceEmotion_ID. Este modelo consigue unos resultados de 66% de acierto sobre dicha base de datos (según sus autores el ojo humano acierta sobre el 65% al clasificar siete emociones). Es capaz de distinguir entre siete emociones diferentes: enfado, asco, miedo, felicidad, tristeza, sorpresa y neutralidad. Devuelve una lista con los porcentajes de posibilidad de cada emoción respecto a la imagen dada.

En la Sección 5.3 se explicará como adaptó este modelo al proyecto.

3.5 Google SpeechRecognition

Por último, como complemento al reconocimiento de emociones en el rostro, se ha incluido también el reconocimiento de emociones en la voz. Para ello fue necesario utilizar el soporte de SpeechRecognition [25], una librería de Python que ofrece diferentes sistemas de

reconocimiento de palabras, como CMU Sphinx, Microsoft Bing Speech, o Google Web Speech API. Para este proyecto se seleccionó este último ya que era el único que no requería de autenticación, aunque se necesita conexión a internet para que funcione. Para su funcionamiento es necesario tener conexión a Internet.

El uso de esta librería es simple. En el siguiente script (en Python3) se enseña un ejemplo de uso de este sistema.

```
1 import speech_recognition as sr
2
3 AUDIO_FILE = ("Emotion/audio.wav")
4
5 # Inicializar Speech Recognition
6 r = sr.Recognizer()
7
8 with sr.AudioFile(AUDIO_FILE) as source:
9     # Leer fichero de audio
10    audio = r.record(source)
11
12 try:
13     # Detectar el discurso que se encuentra en el audio
14     # seleccionando el reconocimiento de palabras de Google.
15     # El idioma escogido es el español.
16     oido = r.recognize_google(audio, language="es-ES")
17     print("El fichero de audio contiene "+oido)
18 except sr.UnknownValueError:
19     print("Google Speech Recognition no pudo entender el audio")
20 except sr.RequestError as e:
21     print("No se pudo solicitar los resultados del servicio Google
22           Speech Recognition; {0}".format(e))
23
24 # Guardar cada palabra detectada por separado en una lista
25 palabras = oido.split(" ")
```

Esta librería se utilizará en el reconocimiento de emociones por audio, el cuál se explicará en la Sección 5.5.

Metodología

LA metodología seguida en este TFG es la definida en el Proceso Unificado de Desarrollo de Software. Su principal característica es el enfoque en los casos de uso y en la arquitectura, permitiendo el refinamiento progresivo de ésta. Esta metodología propone el desarrollo de un esquema iterativo e incremental. Las distintas iteraciones se centran en aspectos relevantes del software, siendo su resultado un sistema ejecutable (a pesar de no estar terminado). En cada iteración se lleva a cabo las fases clásicas de análisis, diseño, implementación y pruebas, de manera que al final del proceso se detecten debilidades, problemas de rendimiento que sean necesarios abordar en la siguiente iteración. Cada iteración incorporará más funcionalidades sobre la anterior, hasta que la última termine con un software que implemente la totalidad de sus funcionalidades.

La ventaja de esta metodología es que es flexible y extensible, con la posibilidad de adaptarse a cualquier tipo de proyectos. Además, al ser iterativa e incremental permite establecer y refinar requisitos para la siguiente iteración. Esto es ideal para este proyecto ya que, al ser de investigación, los objetivos de una iteración dependerán por una parte de los resultados conseguidos en la iteración previa.

4.1 Planificación y gestión del proyecto

4.1.1 Planificación

Como se puede ver en el diagrama de Gantt de la Figura 4.1, el desarrollo del proyecto se dividió en cuatro iteraciones.

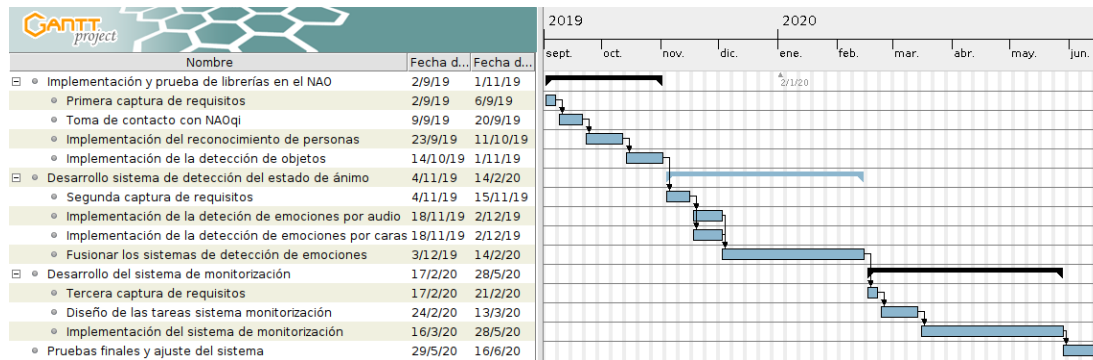


Figura 4.1: Diagrama de Gantt de la planificación del proyecto.

Iteración 1: Implementación y pruebas de librerías en el NAO

- Fecha de inicio: 2 de septiembre de 2019.
- Fecha de finalización: 1 de noviembre de 2019.
- Objetivo: Probar las funcionalidades del robot con la API NAOqi. Probar y adaptar para el NAO los sistemas de detección de personas (FaceNet) y objetos (YOLO) para su futuro uso en SARDAM.
- Tareas a realizar: Primera captura de requisitos. Toma de contacto con NAOqi. Implementación del reconocimiento de personas. Implementación de la detección de objetos.

Iteración 2: Desarrollo del sistema de detección del estado de ánimo

- Fecha de inicio: 4 de noviembre de 2019.
- Fecha de finalización: 14 de febrero de 2020.
- Objetivo: Desarrollar un sistema que detecte el estado de ánimo del usuario, fusionando otras implementaciones que detectan por imagen o por audio para lograr una mayor exactitud en los resultados respecto a los que dan cada uno por sí solos.
- Tareas a realizar: Segunda captura de requisitos. Implementación de la detección emociones por imagen. Implementación de la detección de emociones por audio. Fusionar los sistemas de detección de emociones.

Iteración 3: Desarrollo del sistema de monitorización

- Fecha de inicio: 17 de febrero de 2020.
- Fecha de finalización: 28 de mayo de 2020.

- **Objetivo:** Diseñar e implementar el conjunto de posibles tareas que formen parte de un horario de la vida diaria de una persona de edad avanzada. Desarrollar el sistema de monitorización de las tareas diarias del usuario haciendo uso de los módulos de NAOqi, de los detectores y del sistema de detección del estado de ánimo implementados en la iteraciones anteriores.
- **Tareas a realizar:** Tercera captura de requisitos. Diseño de las tareas del sistema monitorización. Implementación del sistema de monitorización.

Iteración 4: Pruebas finales y ajuste del sistema

- **Fecha de inicio:** 29 de mayo de 2020.
- **Fecha de finalización:** 16 de junio de 2020.
- **Objetivo:** Comprobar que el sistema funciona correctamente y que obtiene una respuesta positiva por parte del usuario a través de una serie de pruebas sobre uno o dos posibles escenarios.
- **Tareas a realizar:** Pruebas finales y ajuste del sistema.

4.1.2 Recursos

Los recursos materiales utilizados son el ordenador portátil donde se programó el proyecto y el robot NAO.

Los recursos humanos son el equipo de desarrollo y los investigadores/directores.

4.1.3 Gestión de riesgos

En la gestión de riesgos se pretende identificar, estudiar y eliminar los posibles factores de riesgo antes de que empiecen a amenazar la finalización satisfactoria de un de un proyecto. Un riesgo es un evento que podría reducir la capacidad para lograr los objetivos definidos en el proyecto.

En la Tabla 4.1 se pueden ver los riesgos que se identificaron al inicio del proyecto, donde también se refleja la probabilidad de ocurrencia del riesgo, el impacto (magnitud de pérdida) que podría producir si se convierten en problema, y la exposición al riesgo.

Dado que se tratar de prevenir y controlar, en la medida de lo posible, la aparición de estos riesgos, se establecieron los planes siguientes que se tuvieron cuenta a lo largo del proyecto.

Los riesgos que se controlarán son **R1, R2, R6 y R7**. Para R1 se analizará la documentación disponible (en el caso del robot) y se evitarán acciones que puedan dañar al hardware. En R2 se analizará el uso de este software y de otras posibles alternativas que lo puedan sustituir. Para

Id	Riesgo	Probabilidad	Impacto	Exposición
R1	Problemas con el hardware utilizado (NAO, ordenador)	Media	Alto	Alta
R2	Problemas con el software utilizado (librerías externas, etc.)	Media	Medio	Media
R3	Captura de requisitos errónea	Media	Alto	Alta
R4	Diseño erróneo del sistema	Media	Alto	Alta
R5	Problemas durante el desarrollo del sistema de detección del estado anímico	Alta	Medio	Alta
R6	Problemas durante el desarrollo del sistema de monitorización de tareas	Media	Medio	Media
R7	Problemas durante la etapa de pruebas	Media	Bajo	Media
R8	Mala planificación del proyecto	Media	Alto	Alta
R9	Recepción negativa del prototipo por parte experto (en este caso, la psicóloga y el personal del centro de mayores)	Baja	Alto	Media

Tabla 4.1: Riesgos capturados en el proyecto.

controlar R6 se empezará implementando desde lo más simple posible, añadiendo complejidad al sistema de manera escalar. También se intentará que el código esté lo más estructurado y modular posible, de manera que sea más fácil corregirlo y/o tomar otras alternativas en casos concretos. Para minimizar el riesgo R7 se harán una serie de pruebas durante el desarrollo para corregir posibles fallos que se puedan dar durante la iteración de las pruebas del sistema.

Los riesgos **R3**, **R4**, **R5**, **R8** y **R9** se tratarán de prevenir al ser de alta exposición. Una captura de requisitos errónea (R3) podría conllevar a tener un sistema resultante que no era el esperado por el cliente. Para evitarlo se analizará exhaustivamente toda la información proporcionada por la experta. Respecto a R4, se analizará que cumpla con los requisitos especificados previamente. Al ser un proyecto de investigación, R5 es un riesgo muy complicado de prevenir, por lo que se diseñarán otras alternativas en caso de que la inicial no sea posible. El riesgo R8 también es difícil de evitar su aparición por la misma razón que en R5. Se analizará en cada iteración la posible duración de ésta, considerando todos los riesgos que puedan ocurrir y estimando una duración más larga de lo natural. Por último, R9, se tratará de prevenir realizando las entrevistas necesarias con el experto y consultando su opinión sobre las diferentes funcionalidades que se pueden agregar al sistema en cada iteración de la planificación.

4.1.4 Estimación de costes

Por último, se hace una estimación del posible coste del proyecto, para lo que se tiene en cuenta los recursos humanos y los materiales. La Tabla 4.2 muestra la evaluación resultante. El precio del ordenador es el que en su día se pagó por él; mientras que el del NAO es el proporcionado por AliveRobots [26]. No se tienen en cuenta posibles amortizaciones.

Respecto a los recursos humanos, según Indeed, un desarrollador/a junior suele cobrar sobre unos 20769€ al año [27], que son unos 1730,75€ al mes, 10,65€/hora. En este proyecto se trabajaron aproximadamente unas 550 horas, por lo que el coste final por la desarrolladora serían unos 5857,50€.

	Nombre	Coste
Recursos materiales	Ordenador portátil ASUS F555L i5-5200U 4GB RAM	500,00€
	NAO Evolution V5	5790,00€
Recursos humanos	Desarrolladora junior	5857,50€
	COSTE TOTAL	12147,50€

Tabla 4.2: Evaluación de costes.

Capítulo 5

Desarrollo

EN este capítulo se explicará cómo se desarrolló SARDAM. Se empezará por la captura de requisitos a cumplir en proyecto para llegar a la arquitectura de SARDAM y sus distintos componentes, que serán explicados en las secciones que componen este capítulo. También se explicará como se diseñaron las tareas que ejecutará SARDAM.

5.1 Captura de requisitos

La captura de requisitos se hizo en tres etapas a medida que avanzaba el proyecto y se necesitaba establecer nuevas funcionalidades. A partir de la información obtenida se pudieron establecer los requisitos finales del sistema.

5.1.1 Primera captura de requisitos: Primera reunión con los investigadores

Los investigadores y la desarrolladora se reúnen por primera vez para establecer la finalidad del proyecto y unos requisitos iniciales a cumplir en esta primera fase. Se definió que se iba a diseñar un sistema SAR dirigido a la población de tercera edad, utilizando el robot NAO y programándolo en Python 2.7.x (salvo en casos excepcionales donde no es posible su uso).

Para empezar se necesitaría de reconocimiento de personas, detección de objetos y reconocimiento de voz, como mínimo. El robot no iba a tener que desplazarse, como mucho mover los brazos y la cabeza; en cambio, sí necesitaba poder comunicarse con el usuario para poder hacer la interacción humano-robot.

5.1.2 Segunda captura de requisitos: Primera entrevista con la experta

Para poder pasar a la siguiente fase del proyecto se necesitaba obtener información de un profesional en la atención a personas mayores. Por lo tanto, se acudió al centro de mayores SUACASA (en San Sadurni) para reunirse con la psicóloga Rocío Romero López y conocer

las posibles necesidades de la población anciana en las que un robot podría ser de ayuda. En este proyecto fue quién aportó el conocimiento experto.

La psicóloga consideró que lo mejor era enfocar el proyecto a las personas mayores que tengan un cierto grado de autonomía pero, por ejemplo, sufran de pérdidas de memoria leves. Los residentes de centros de mayores (que normalmente son totalmente dependientes) o con un deterioro cognitivo más grave serían más complicadas de tratar, como los que padecen de Alzheimer. Normalmente, son personas inexpresivas y suelen tener conductas agresivas e inesperadas, factores que complicarían la interacción humano-robot. Además, como se comentó en la introducción (Sección 1.1), también se enfatizó en evitar el ingreso en residencias a aquellas personas que aún podían valerse por sí mismas con un poco de ayuda y seguimiento.

La experta enfatizó la importancia de que la persona cumpliera con una rutina diaria, tanto si es en el centro de mayores como en su hogar. Si es necesario, se le animaría a que la llevara a cabo en caso de que se niegue a hacerlo. A partir de esta información se estableció un nuevo objetivo del proyecto: dotar al sistema de la capacidad del sistema de detectar el estado de ánimo del usuario, con el objetivo de adaptar la interacción y así lograr la realización de las tareas diarias.

Respecto a qué otras actividades se podían hacer con una persona mayor, la experta recomendó los juegos cognitivos y actividades de reminiscencia, que estimulan la memoria y evitan el deterioro de la mente. También comentó que a los ancianos les gusta mucho la música y les suele levantar el ánimo. Una actividad que se lleva a cabo en la residencia es la denominada como "Libro de vida", en el que se va recopilando poco a poco datos de la vida pasada de cada residente, como cuándo y dónde nació, fotografías antiguas, eventos importantes de su vida, etc. Se iba haciendo con la ayuda de los familiares y de las cuidadoras, ya que los residentes de este centro estaban en muy mal estado cognitivo (muchos de ellos tenían Alzheimer). Otra opción parecida era la de mantener una caja pequeña con objetos que les recuerde a eventos de su pasado, aunque está más enfocada para personas con pérdidas de memoria más graves. Esta información resultó útil a la hora de diseñar las tareas, ya que este conocimiento práctico y esta experiencia enriquecen en gran medida el sistema de monitorización.

5.1.3 Tercera captura de requisitos: Segunda reunión con la experta

Después de crear un sistema de detección del estado de ánimo, se necesitaba saber más sobre la rutina diaria de una persona de edad avanzada, y cómo hacer uso del estado de ánimo para favorecer el cumplimiento de esta rutina y poder diseñar el sistema de monitorización. Por estos motivos, se realizó otra entrevista a la psicóloga del centro de mayores.

Como en la reunión anterior, la experta recalcó la importancia de que la persona mantenga una rutina diaria con hábitos saludables, y la necesidad de que se sienta útil, que es uno de los factores que más felicidad les aporta a este colectivo. Respecto a dicha rutina, propor-

cionó una serie de documentos que se deberían tener en cuenta a la hora de realizar el diseño de dicho sistema. En ellos se trataba el tema de las actividades de la vida diaria (AVD) para una persona de la tercera edad (véase Apéndice A). Las AVD son un conjunto de tareas que componen la vida diaria de una persona, permitiendo que viva de manera autónoma e integrada en su entorno. La psicóloga consideró que sería interesante controlar que las cumplen, especialmente aquellas tareas que son básicas para su autonomía.

Se le comentó sobre el sistema de detección del estado de ánimo, y le pareció buena idea el tratar de levantar el ánimo de la persona si expresa que no está de humor para hacer alguna de las actividades. Primero se podría intentar convencer de que lo haga, y si no se da el caso entonces hacer que se entretenga con otra cosa mientras no se encuentre mejor para ello, o cambiarla por otra actividad posterior que prefiera hacer antes que la actual. Esas actividades con las que se podría mejorar el estado anímico de la persona podrían ser juegos cognitivos o actividades de reminiscencia (de los que ya se hablaron en la entrevista previa), o ejercicio de movilidad física para la población de edad avanzada, denominado como gerontogimnasia.

A partir de la información dada por la psicóloga, el equipo de desarrollo y los investigadores se reunieron para determinar las características y funcionamiento de SARDAM. Se definió que SARDAM, en una primera versión, será un sistema de monitorización de la vida diaria de una persona de edad avanzada que vive sola, utilizando la detección del estado de ánimo para controlar que cumple con su rutina, y si es necesario tratar de motivarla con alguna otra actividad en caso de posible incumplimiento del horario. El objetivo de este sistema es el bienestar de la persona mayor, manteniéndola ocupada, fomentando el sentimiento de utilidad y estableciendo hábitos saludables en su día a día para aumentar su calidad de vida. Además, el sistema permite llevar un registro de las actividades realizadas y de las no realizadas. Esta información puede ser de gran utilidad para los cuidadores, médicos, e incluso por la propia familia.

El diseño de las tareas desarrolladas se basó en las AVD. Según los documentos proporcionadas por la experta, cada una de estas actividades trabaja con las habilidades y destrezas fundamentales de la persona. Si no es capaz de llevarlas a cabo independientemente puede significar que tenga algún déficit cognitivo-conductual y/o motor.

De dichos documentos también se extrajo la clasificación de dichas actividades, que resultó útil a la hora de asignar prioridades a las tareas. Se establecen en tres grupos diferentes: actividades básicas de la vida diaria (ABVD), actividades instrumentales de la vida diaria (AIVD), y actividades avanzadas de la vida diaria (AAVD). Esta clasificación se comenta en más profundidad en la Sección 5.6.1.

En esta reunión se establecieron los requisitos finales del proyecto, que se describen a continuación.

5.1.4 Requisitos funcionales

Los requisitos funcionales son aquellos que definen cómo debe comportarse el sistema. Con la información recopilada hasta el momento, se determinaron los siguientes requisitos funcionales. Por parte de la experta son:

- El sistema tiene que ejecutar una serie de tareas que componen la rutina diaria del usuario preferentemente en el orden establecido por el horario.
- Las tareas seguirán las características definidas en el documento sobre las actividades de la vida diaria (véase Apéndice A, tercer documento).
- La ejecución de una tarea dependerá de si el estado de ánimo del individuo es apto para su realización.
- Debe existir un conjunto de actividades que sirvan para levantar el ánimo al usuario en caso de que sea necesario para que pueda realizar la tarea.
- El sistema tiene que tratar que el usuario haga las tareas propuestas, especialmente las que están relacionadas con las ABVD; las AAVD no tienen que controlarse tanto al no ser determinantes para la vida autónoma.
- El sistema debe mantener un seguimiento de la realización de las tareas, guardando el estado emocional de la persona para cada una de las tareas realizadas.

Los requisitos definidos por los investigadores son:

- El horario debe estar descrito en un fichero de configuración, indicando la hora de inicio del día y el conjunto de tareas con sus respectivas propiedades.
- Cada tarea tiene asignada una prioridad, una duración y un conjunto de estados de ánimo que indican los estados que propician que la persona realice las acciones de la tarea.
- El sistema debe permitir añadir, eliminar y modificar fácilmente las tareas.
- El sistema solo será utilizado por un usuario en concreto.

5.1.5 Requisitos no funcionales

Los requisitos no funcionales, establecidos por los investigadores, son:

- El dispositivo hardware a utilizar es el robot NAO.

- El lenguaje de programación que se debe usar en el proyecto es Python 2.7.x, excepto en los casos en los que no sea posible.
- El sistema debe ser modular. Facilitando que se puedan añadir nuevas funcionalidades sin afectar demasiado a otras partes del sistema.
- El sistema tiene que ser reutilizable.
- El sistema debe ser mantenible.

5.2 Arquitectura de SARDAM

La arquitectura propuesta representa el prototipo de un sistema SAR, centrado en la monitorización de la vida diaria de una persona de edad avanzada, basándose en el estado de ánimo de dicha persona para la realización de cada actividad.

Como se puede observar en la Figura 5.1, la arquitectura está compuesta por cinco bloques: el robot NAO, los detectores de información sensorial, el detector del estado de ánimo, el sistema de monitorización de tareas y los actuadores.

Comenzando por la parte superior, el NAO interactúa directamente con el usuario. A partir de esta interacción, el robot ofrece a los detectores la información que obtienen sus sensores, y que será posteriormente procesada para dejarla disponible a estos módulos. Tanto el detector del estado de ánimo como el sistema de monitorización tienen acceso a esos datos generados por los detectores. El detector del estado de ánimo predice cual es el estado de ánimo del usuario y se lo ofrece al sistema de monitorización cada vez que lo necesite. Respecto al sistema de monitorización, se encarga de cargar y ejecutar el horario que contiene las tareas a realizar diariamente, el cual se obtiene de un fichero de configuración. Este fichero está escrito, idealmente, por un especialista, como el médico de cabecera del usuario, ya que conoce su condición física y mental y sabe que es lo que necesita para seguir manteniendo un ritmo de vida saludable. Además, guardará la información recopilada sobre la monitorización de la actividad diaria del usuario generada en un fichero log, y que puede ser accedida por dicho especialista. Durante la ejecución de las tareas, el sistema de monitorización da órdenes a los actuadores del robot de lo que tienen que hacer dependiendo de las acciones de interacción que se tengan que realizar. Estos actuadores se encargarán de que el robot se comporte como es esperado.

En las siguientes secciones se explicará más detalladamente el desarrollo realizado en cada uno de estos bloques, menos el del NAO, que ya se describió en la Sección 3.1 de Fundamentos tecnológicos.

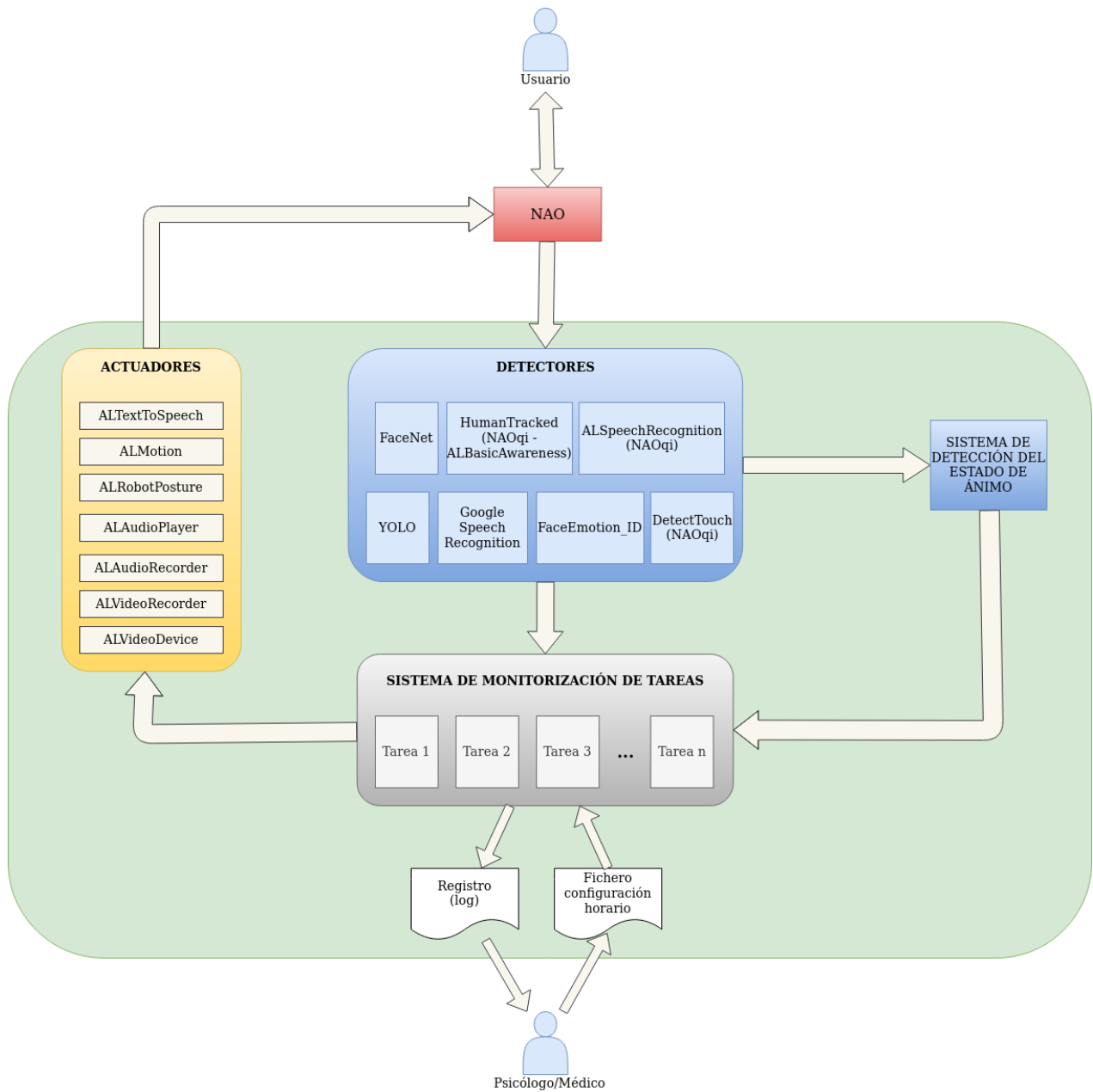


Figura 5.1: Arquitectura del sistema SARDAM.

5.3 Detectores

Los detectores son los que se encargan de recopilar la información que reciben de los sensores del NAO y procesarla para que posteriormente la utilicen el resto de los módulos. SARDAM dispone de siete detectores diferentes en esta primera versión. Como se puede ver en la Figura 5.1 son *ALSpeechRecognition*, *HumanTracked*, *DetectTouch*, *FaceNet*, *YOLO*, *FaceEmotion_ID* y *Google SpeechRecognition*.

ALSpeechRecognition es un módulo perteneciente a NAOqi que sirve para detectar un conjunto de palabras, predefinido previamente, en el discurso del usuario.

HumanTracked y *DetectTouch* son dos módulos nuevos que se definieron para poder suscribirse a ciertos eventos del NAO. *HumanTracked* detecta si hay personas en la visión del robot, mientras que *DetectTouch* detecta cuando una persona toca al robot. Para su creación se hace uso de los módulos correspondientes de NAOqi (*ALModule*, *ALBroker*, *ALMemory*). En el caso de *HumanTracked*, se necesita de los eventos “*HumanTracked*” y “*HumanLost*” de *ALBasicAwareness* para poder saber si se encuentra o no alguna persona en el campo de visión del robot. Para *DetectTouch* se utiliza el evento “*TouchChanged*” de *ALTouch* para detectar si el usuario toca ciertas partes del robot (Figura 5.2) e informa cuáles son.

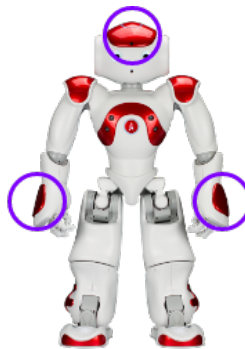


Figura 5.2: Zonas del NAO dónde se debe tocar para *DectTouch*.

El resto de detectores no están basados en NAOqi; son los mencionados en Fundamentos tecnológicos: *FaceNet* (Sección 3.2), *YOLO* (Sección 3.3), *FaceEmotion_ID* (Sección 3.4) y *Google SpeechRecognition* (Sección 3.5). Estos detectores fueron implementados en Python3 por incompatibilidad con Python2, excepto *Google SpeechRecognition*, que fue añadido como funcionalidad en Python2, pero en el sistema de detección de emociones se utilizó Python3(para evitar los problemas con unicode que tenía la versión de Python2).

Para la integración de *FaceNet* nos basamos en el tutorial de *How to Develop a Face Recognition System Using FaceNet in Keras* [20]. Para la detección de la cara se hizo uso del método descrito en [28], debido a su buen funcionamiento, además que no implica tener en cuenta el

ángulo con el que se detecta la cara (si está girada o no). La implementación del vector en el NAO es sacar imágenes, sacar *face-embedding*, entrenar con ella. Una vez detectada la cara, se extrae el rostro en una imagen de 160x160, que servirá como entrada al modelo para obtener el *face embedding*. Para el reconocimiento de la persona se hace una clasificación con SVM de todos los *face embeddings* de la base de datos mas el de la imagen a reconocer. En caso de que solo hay una persona en la base de datos no se puede hacer una clasificación con SVM, por lo que se calcula la media de la distancia euclídea del vector de la imagen recibida respecto al conjunto de entrenamiento. Si no supera un umbral de 1.1 entonces es la misma persona (0.0 significa que los rostros son idénticos, y 4.0 que son totalmente diferentes). Dicho umbral es mencionado en el documento como suficiente para una correcta clasificación de las imágenes.

Para integrar YOLO al proyecto se siguió el tutorial *How to Perform Object Detection With YOLOv3 in Keras* [22]. La diferencia con lo implementado en el tutorial es que guarda las etiquetas resultantes en un fichero numpy comprimido (*objects.npz*), ya que este código está escrito en Python3. Así se pueden acceder a los resultados desde las otras partes del sistema que utilizan Python2.

Al igual que con FaceNet, se adaptó FaceEmotion_ID para que utilizara el mismo método de detección de caras (el descrito en [28]). También se agruparon las siete emociones que detecta (enfado, asco, miedo, felicidad, tristeza, sorpresa y neutralidad) para formar tres tipos de estado de ánimo: animado, normal y desanimado. Este paso se explicará más a fondo en la Sección 5.5.

5.4 Actuadores

Los actuadores son los que ejecutan las acciones necesarias en el robot. En SARDAM se utilizan seis módulos de NAOqi que permiten que el robot hable (ALTextToSpeech), se mueva (ALMotion) o se coloque en una posición específica (ALRobotPosture), emita audios (ALAudioPlayer), grabe audios (ALAudioRecorder) y vídeos (ALVideoRecorder), y obtenga imágenes de las cámaras del robot (ALVideoDevice). Estos módulos se describieron en la sección 3.1.1 de Fundamentos tecnológicos.

5.5 Detección del estado de ánimo

Como se estableció previamente en los requisitos, SARDAM gestiona las tareas a proponer en base a unas prioridades, y al estado de ánimo del usuario. Es decir, si el usuario no se encuentra bien para realizar una cierta tarea, entonces el robot intentará motivarlo para que lo haga de forma diferente en base al estado detectado. Para poder realizar esto, se necesitó desarrollar un sistema de detección del estado de ánimo.

Se pretendía tener un detector robusto. Para ello se decidió utilizar más de un tipo de sensor para captar emociones, es decir, obtener la emoción por varias vías. De esta manera los resultados se basarían en más de un solo aspecto. En este caso se escogieron sistemas que utilizaban la imagen y el audio, fusionándolos para tener un sistema de detección de emociones híbrido.

Durante el desarrollo se tuvieron que realizar tres aproximaciones para la fusión de los dos sistemas de detección.

Primera aproximación

En un principio se quería detectar las emociones por su imagen facial, con FaceEmotion_ID, y por el tono de voz de una persona. Para este último se probaron varios detectores (Open Vokaturi [29], Speech Emotion Recognition [30]), pero el que obtuvo mejores resultados (con poca diferencia) fue Speech Emotion Analyzer [31], que fue el escogido para esta aproximación.

Debido a que FaceEmotion_ID detecta siete emociones (enfado, asco, miedo, felicidad, tristeza, sorpresa y neutral) pero Speech Emotion Recognition cinco (enfado, calma, miedo, felicidad y tristeza), se tuvo que adaptar FaceEmotion_ID para que devolviera las mismas cinco emociones que el de audio. Por lo tanto, las emociones asco, sorpresa y miedo se consideraron como una sola: miedo. Esto es posible por ciertas características que comparten, descritas en el capítulo 5 del libro *Psicología de la emoción* [32]. En él se menciona la relación entre el asco y el miedo por los vínculos que puede tener la primera a estados fóbicos, además de tener otros aspectos en común. También la sorpresa y el miedo son muy parecidos en expresiones faciales (son las que comparten más unidades de acción), y se tienden a confundir. Además de lo anterior, también se consideró que calma y neutral significarían lo mismo. En ambos casos, tanto el detector de emociones como el detector por imagen, devuelven la probabilidad de cada una de las cinco etiquetas semánticas que califican una expresión.

Para fusionar las salidas de los dos detectores se utilizaron los tres métodos que W. Hashim Abdulsalam et al expusieron en su artículo sobre un sistema de reconocimiento emocional basado en técnicas híbridas [33]. El primero de los métodos es el **majority vote**, con el que se asume como etiqueta final para calificar la emoción la que obtiene el mayor valor entre los resultados del audio y vídeo. El segundo es hacer la **media** de la suma de las precisiones de audio y vídeo de cada emoción, seleccionando la de mayor valor. Por último está la **ponderación**, donde para cada emoción se multiplican los resultados de vídeo por 0.6 y los de voz por 0.4, y se suman; después se selecciona el de mayor valor. Dichos pesos son los recomendados por el artículo citado.

Se realizaron varias pruebas comparando los tres métodos para investigar cual de los tres era el que mejores resultados obtenía. Para ello, se eligieron una serie de vídeos con sus res-

pectivos audios de la base de datos de emociones RAVDESS [34], y se detectaron las emociones por rostro y por tono de voz.

La Tabla 5.1 muestra los resultados de las pruebas que se realizaron. En dicha tabla, las columnas *Majority vote*, Media y Ponderación muestran la etiqueta resultante de cada uno de los métodos de mezcla junto a la probabilidad asociada expresada en %. Se escogieron 16 vídeos de varios actores, de los que se utilizaron 13 frames de cada uno para la detección por imagen (se obtiene la emoción final a partir del promedio de los resultados de dichos frames). Como se puede observar, *majority vote* es, en general, el que obtiene los resultados más altos (en cuanto a la probabilidad de la emoción), pero la ponderación es el método que más acierta si comparamos la etiqueta resultante con la emoción real. También se puede observar que la mayor parte de los resultados finales se corresponden con la emoción real, pero la emoción por imagen ha fallado en casi todos los casos en los que la emoción era negativa, mientras que la neutralidad y la felicidad las acertaba siempre. Se puede decir que FaceEmotion_ID funciona bien menos en los casos de que sean emociones negativas. En otras pruebas que se realizaron, con diferente número de frames, los resultados no variaban mucho a los de esta prueba.

Cabe destacar que en *majority vote* siempre se escogían los resultados de la detección por audio, ya que siempre eran los más altos (los porcentajes estaban sobre el 90%). Esto podía ser debido a que Speech Emotion Recognition fue entrenado con RAVDESS, que era la misma base de datos que la de las pruebas. Además, los actores de los vídeos de dicha base de datos no eran muy expresivos facialmente, por lo que podía ser la razón por la que la detección de emociones por rostro no fuera tan buena. Por lo tanto, no se pudo tomar estos resultados como lo suficientemente fiables como para determinar la fusión como válida.

El siguiente paso fue probarlo con vídeos caseros, de manera que se pudiera verificar un correcto funcionamiento del sistema híbrido al que se quería llegar en esta aproximación. Estos vídeos, de corta duración (sobre tres segundos), consistían en decir una frase que representara cada una de las emociones, como “No me encuentro bien, estoy triste” o “Estoy muy enfadada”, y tratando de que las expresiones faciales fueran claras. Pero, en las pruebas que se realizaron con ellos, los resultados fueron malos, Speech Emotion Analyzer no era capaz de detectar bien alguna de las emociones.

Al ver que no se conseguía mejorar el sistema, se decidió agrupar las emociones en tres tipos de estado de ánimo que se consideraron suficientes para el ámbito de aplicación de este proyecto. Estos grupos eran **Animado** (felicidad), **Normal** (neutral), **Desanimado** (enfado, tristeza, miedo). Se probó individualmente cada detector para ver si mejoraban sus predicciones. Los resultados de detección por el tono de voz siguieron siendo malos, mientras que FaceEmotion_ID conseguía mejores resultados, especialmente en los casos de buen estado de ánimo.

Actores	Etiqueta vídeo	Etiqueta audio	Majority vote (Prob%)	Media (Prob%)	Ponderación (Prob%)	Real
1	asqueada	feliz	feliz (87,62)	feliz (57,28)	miedoso (78,19)	miedoso / asqueado
2	neutral	miedoso	miedoso (70,32)	neutral / calmado (40,89)	neutral / calmado (46,17)	calmado / neutral
3	neutral	calmada	calmada (99,93)	neutral / calmado (40,89)	neutral / calmado (46,17)	calmado / neutral
4	neutral	miedosa	miedosa (99,88)	miedoso (81,82)	miedoso (78,20)	miedoso
5	neutral	calmada	calmada (98,33)	neutral / calmado (40,89)	neutral / calmado (46,17)	calmado / neutral
6	neutral	miedosa	miedosa (99,0026)	miedoso (81,82)	miedoso (78,20)	miedoso
7	neutral	calmado	calmado (83,099)	neutral / calmado (40,89)	neutral / calmado (46,17)	calmado / neutral
8	neutral	triste	triste (99,36)	triste (74,28)	triste (69,47)	miedoso / sorprendido
9	neutral	enfadado	enfadado (99,68)	enfadado (45,46)	enfadado (41,23)	enfadado
10	feliz	feliz	feliz (93,48)	feliz (57,28)	feliz (48,76)	feliz
11	feliz	triste	triste (99,98)	triste (74,28)	triste (69,47)	triste
12	neutral	triste	triste (98,74)	triste (74,28)	triste (69,47)	triste
13	feliz	feliz	feliz (98,66)	feliz (57,28)	feliz (48,76)	feliz
14	feliz	miedosa	miedosa (99,13)	miedoso (81,82)	miedoso (78,20)	miedoso / sorprendido
15	neutral	triste	triste (82,89)	triste (74,28)	triste (69,47)	miedoso / asqueado
16	asqueada	enfadado	enfadado (99,40)	enfadado (45,46)	enfadado (41,23)	enfadado

Tabla 5.1: Resultados de la fusión de detección de emociones por rostro y tono de voz.

Al final se descartó la detección por tono de voz, y se replanteó como abordar el problema.

Segunda aproximación

En esta aproximación se reutilizó y adaptó la versión del sistema de detección de emociones por rostro en la que se devolvían tres estados de ánimo, desarrollada en la primera aproximación. Como la detección por tono de voz resultó ser mucho peor de lo esperado, se decidió probar con detectar palabras que se puedan asociar a un cierto estado de ánimo en el habla de una persona. Esta nueva aproximación se implementó con Google SpeechRecognition.

Para la implementación del sistema de detección del estado de ánimo por el habla, primero se definió un vocabulario de posibles palabras que se pueden dar en una respuesta a un tipo de preguntas cerradas que impliquen el estado emocional de una persona (“¿Qué tal estás?”, “¿Cómo te encuentras?”). Dicho vocabulario se clasifica en tres estados de ánimo:

- Animado: feliz, encantado, encantada, alegre, bien, buen, genial, contento, contenta, animado, animada, ganas, energía, energías, quiero, apetece, sí, gusta.
- Normal: regular, normal, bah, va, tirando, bueno, tranquilo, tranquila, ni, sé, vale.
- Desanimado: cansado, cansada, desganado, desganada, agotado, agotada, triste, enfadado, enfadada, cabreado, cabreada, malo, mala, mal, muriendo, muero, enfermo, enferma, exhausto, exhausta, fatal, harto, harta, irritado, irritada, duele, doler, sin, no.

El funcionamiento de este sistema es el siguiente. Primero se obtiene el texto de lo que dijo el usuario en el audio con Google SpeechRecognition, guardando cada palabra individualmente en una lista. Después se comprueba a qué estado de ánimo pertenece cada palabra de dicha lista, aumentando en uno el contador de dicho estado y el contador de las palabras encontradas del vocabulario, y añadiendo la palabra a una lista donde se guardan las palabras del vocabulario encontradas. Para saber a qué estado pertenece cada palabra también se tiene en cuenta si antes de ella apareció otra palabra que cambie su significado. Para abordar este problema se hizo de esta manera:

- Si se detecta una palabra relacionada con buen estado de ánimo (Animado), pero la palabra anterior es “no” o “sin”, entonces se suma uno más al contador del mal estado en vez de al bueno. En cambio, si es “ni” se aumenta el de estado neutral. Por ejemplo, la frase es “No me encuentro bien” tiene dos palabras del vocabulario, que son “No” y “bien”. Esta última pertenece al buen estado, pero al encontrarse “No” antes que ella, éste le da un significado negativo.

- Si se detecta una palabra relacionada con estado de ánimo neutral (Normal), se tienen en cuenta las palabras “no” y “sé” para así detectar el “No sé”. Se disminuye en uno el contador de desanimado y se aumenta en dos el de neutral.
- Si se detecta una palabra relacionada con mal estado de ánimo (Desanimado) pero la palabra anterior es “no” o “sin”, entonces se disminuye en uno el contador de desanimado (ya que habría sido aumentado cuando se encontraron las anteriores), y se suma dos más al contador de animado. Un ejemplo es cuando la frase es “No estoy agotada”, las palabras detectadas son “no” y “agotada”. La primera anula el valor negativo de la segunda, aportándole un valor positivo. Como en buen estado de ánimo, también si la palabra anterior a la actual es “ni”, entonces su valor se convierte a neutral (por ejemplo, “Ni estoy triste ni estoy feliz”).

Para obtener los porcentajes finales, se divide el número de palabras encontradas en cada estado de ánimo por el número total de palabras encontradas del vocabulario.

```
The audio file contains: sí sí estoy muy bien bien
B sí
B sí
B bien
B bien
Resultado final audio:
Porcentaje ANIMADO: 100.0
Porcentaje NORMAL: 0.0
Porcentaje DESANIMADO: 0.0
```

Figura 5.3: Ejemplo de resultados de la ejecución del sistema de detección de emociones por palabras.

En la Figura 5.3 se puede ver un ejemplo de resultados de la ejecución de este sistema.

Una vez finalizada la implementación de la detección del estado de ánimo por palabras, el siguiente paso era crear un modelo con técnicas de aprendizaje automático donde se fusionarían los dos sistemas de detección. Se tendría en cuenta el tipo de personalidad del usuario. Con esto se pretendía abordar los casos en los que el resultado de cada detector por individual no se correspondía con la emoción actual de la persona. Estos casos fueron representados en seis personalidades básicas que el sistema se podría encontrar:

- **Personalidad 1.** Persona sincera y expresiva. Es fiable tanto en expresiones de la cara como en lo que dice (suelen coincidir).
- **Personalidad 2.** Persona con retranca, bromista. Es más fiable en la expresión que en lo que dice. Cuando está de mal humor no bromea.
- **Personalidad 3.** Persona poco expresiva facialmente pero normalmente honesta. Debemos fiarnos más de lo que dice. Suele contestar con frases cortas.

- **Personalidad 4.** Persona que suele decir que se encuentra bien o que todo está bien o normal, a pesar de que en la cara se le nota que no está tan bien. En esos casos debemos fiarnos más de la expresión. Solo cuando está muy mal diría que se encuentra mal.
- **Personalidad 5.** Persona negativa, siempre se queja, aunque esté de buen humor. Debemos considerar un estado de ánimo más alto de lo que dice excepto cuando se le ve mal.
- **Personalidad 6.** Persona positiva, muy sonriente, independientemente de como se encuentre. Debemos fiarnos más de lo que dice que de la expresión.

Como se ha dicho previamente, se reutilizó la versión adaptada de FaceEmotion_ID de la aproximación anterior. Esta versión devolvía tres estados de ánimo (Animado, Normal, Desanimado), pero se tuvo que ajustar para que tuviera cinco estados de ánimo: Muy animado, Animado, Normal, Desanimado, Muy desanimado. Este paso era necesario para poder generar más datos de entrenamiento para la red. Por lo tanto, si los porcentajes de Animado o Desanimado eran superiores al 70%, pasarían a ser Muy animado o Muy desanimado, ya que simbolizaba que la probabilidad de que fuese esa emoción la detectada era muy alta.

Las etiquetas del detector por el habla eran Animado, Normal y Desanimado. En dicho detector no se clasificó en cinco estados como se hizo con el detector por imágenes porque no se podía determinar con certeza un estado medio entre cada uno de ellos.

Para generar el conjunto sintético de datos que se utilizaría para entrenar la red, se partió de las seis personalidades definidas que se mencionaron previamente. Por cada personalidad se crearon todas las combinaciones posibles entre las etiquetas de audio y de imagen. Dichas combinaciones representan la etiqueta correcta cuando se dan ciertas posibilidades en los resultados de los detectores. Por ejemplo, en la personalidad 6, si se da que está Desanimado por audio pero Muy Animado por vídeo, entonces el resultado debería de ser Desanimado. En total se generaron 56 datos, quedando un conjunto mucho más pequeño de lo esperado.

Por lo tanto, las entradas al modelo serían:

- Una de las cinco etiquetas de la detección por imagen: Muy animado, Animado, Normal, Desanimado y Muy desanimado.
- Una de las tres etiquetas en detección por audio: Animado, Normal y Desanimado.
- El identificador (id) de la personalidad (del 1 al 6).

La salida sería el estado de ánimo resultante de fusionar las dos técnicas con respecto a la personalidad dada. Se darían tres posibles etiquetas: Animado, Normal y Desanimado.

Se implementó la red y se entrenó con el conjunto de datos generado. Pero al final se tuvo que desistir de la idea por los malos resultados de este modelo (0.61 ± 0.22), debido en parte al conjunto de datos de entrenamiento pequeño que no se consiguió mejorar.

Tercera aproximación

En la versión final se terminó utilizando para el sistema híbrido de detección de emociones las implementaciones de detección por vídeo y audio, esta última de la aproximación anterior. En la de vídeo se retomó la detección a tres estados de ánimo implementada en la primera aproximación, para que así tuviera los mismos que la de audio.

En la Figura 5.4 se muestra gráficamente cómo funciona el sistema de detección del estado de ánimo que finalmente se usa en este proyecto. Como se puede observar, la detección del estado de ánimo por audio es de la misma manera que la explicada en la segunda aproximación: clasificar las palabras del audio en los tres estados de ánimo y, a continuación, calcular la probabilidad de cada uno de los estados. Para la detección en el vídeo, cuya duración se espera que sea de 5 segundos, primero se obtienen las emociones de cada uno de los 10 frames a detectar. A continuación se hace el promedio de dichos resultados (todos los frames) por cada una de las siete emociones, para conseguir los resultados finales del vídeo. El siguiente paso es agrupar las emociones en los tres estados de ánimo (como se comentó en la primera aproximación), sumando las probabilidades de las emociones agrupadas.

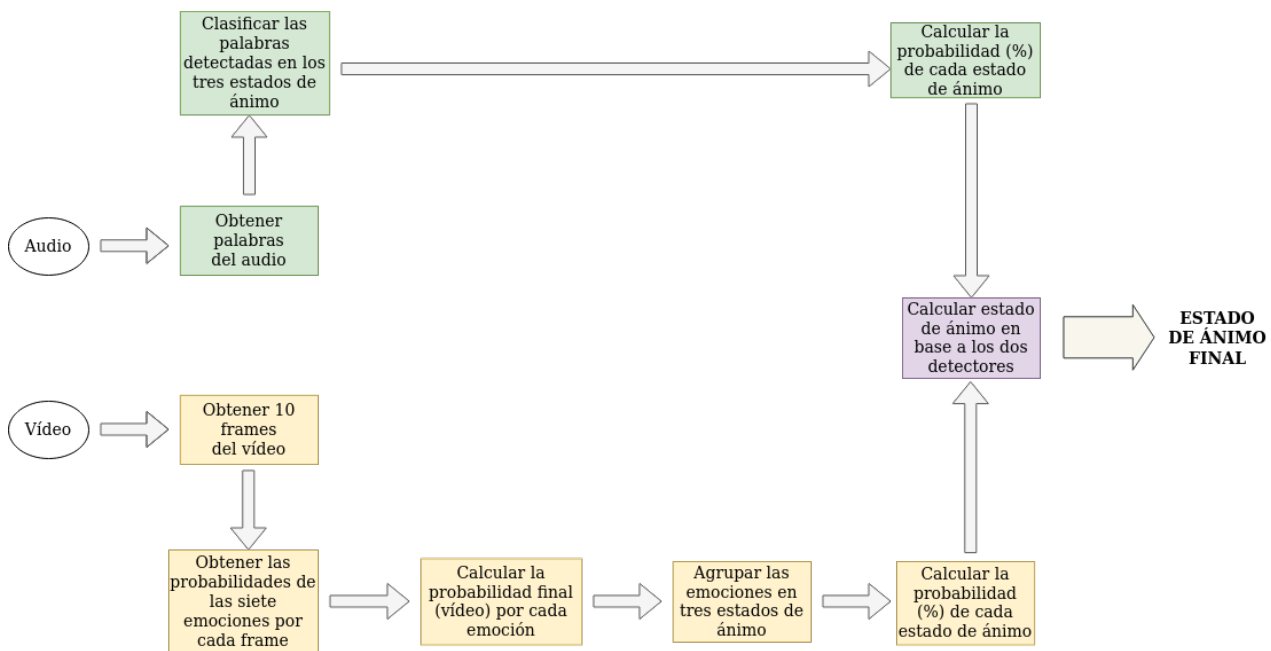


Figura 5.4: Esquema gráfico del funcionamiento del sistema de detección del estado emocional.

Una vez obtenido los tres estados de ánimo con sus probabilidades por cada detector por separado, se calcula la media de la suma de los porcentajes de cada detector por cada estado de ánimo. Esto es, la media de la suma de los porcentajes de Desanimado (DesanimadoVideo+DesanimadoAudio / 2), la media de la suma de los porcentajes de Normal, y la media de

la suma de los porcentajes de Animado. Una vez calculados los porcentajes finales de cada estado de ánimo, se convierten los tres estados de ánimo en los cinco siguientes:

[MUY DESANIMADO, DESANIMADO, NORMAL, ANIMADO, MUY ANIMADO]

Estar Desanimado, por ejemplo, sería como estar cansado o un poco triste, pero no tan mal como si estuviera Muy desanimado. Lo mismo en el caso de Animado y Muy animado. De esta manera los casos Muy desanimado y Muy animado surgirán si los dos detectores están de acuerdo en que es dicho estado de ánimo, siendo un porcentaje alto la probabilidad de que lo sea.

Se definieron unas reglas para convertir los tres estados en los cinco estados:

- Si uno de los estados Desanimado, Normal o Animado superaba el 70% de acierto, entonces es prácticamente seguro que la persona estuviera en los extremos Muy desanimado, Muy animado y Normal.
- Si no, se mira cuál es el segundo estado de ánimo con más porcentaje. Si son Desanimado y Normal las contiguas (da igual cuál de las dos es la de mayor porcentaje), entonces la emoción será Desanimado. Pero si son Animado y Normal la emoción final es Animado. Con Animado y Desanimado no se puede dar porque son estados contrarios.
- Si no se dan los casos de anteriores (si uno es Animado y el otro Desanimado), no se puede detectar la emoción por este sistema ya que hay diferencias entre los dos detectores. En este caso, se escoge el resultado del detector de palabras, ya que es el más fiable.

Una vez hecho estos cálculos, se puede obtener el estado emocional final por las dos técnicas de vídeo y audio. Como se ha dicho previamente, esta es la aproximación que se utilizó en este TFG.

5.6 Sistema de monitorización de tareas

El sistema de monitorización de tareas es el encargado de ejecutar el conjunto de tareas de la rutina diaria del usuario. Es la parte más importante del proyecto, ya que es el que recibe toda la información procesada de los detectores, y la analiza para así dar ciertas órdenes a los actuadores; es el que intercomunica todas las partes de SARDAM.

Después de realizar la tercera captura de requisitos, donde se recopiló información sobre las AVD, se diseñó el conjunto de tareas que formarían parte de la rutina diaria del usuario. Una vez obtenido dicho conjunto se diseñó e implementó todo el sistema SARDAM.

5.6.1 Diseño de tareas

Se empezó seleccionando el conjunto de tareas que formaría parte del plan diario, con la ayuda de la información que proporcionó la psicóloga, especialmente los documentos sobre las actividades de la vida diaria (véase Apéndice A). Como se comentó en la sección 5.1.3, las AVD se clasifican en tres grupos:

- **Actividades básicas de la vida diaria (ABVD).** Son fundamentales para la supervivencia del individuo. Cubren las necesidades básicas que realiza diariamente el ser humano, como alimentarse, asearse, vestirse, dormir, etc. Son las más importantes y determinan si la persona es capaz de llevar una vida autónoma.
- **Actividades instrumentales de la vida diaria (AIVD).** Están más relacionadas con el entorno de la persona. Implican una mayor complejidad cognitiva y motriz. Engloba actividades como comunicarse de diferentes maneras (escribir, hablar por teléfono, etc.), mayor movilidad en su entorno exterior, capacidad de mantenerse saludable y administrarse la medicación, limpiar la casa, o ir a la compra, entre otras.
- **Actividades avanzadas de la vida diaria (AAVD).** No son necesarias para que el individuo lleve una vida independiente. Son tareas relacionadas con el trabajo, la educación, viajar, actividades de ocio, etc., en definitiva con el estilo de vida del individuo.

Teniendo en cuenta esta clasificación se diseñó el plan diario del usuario, agrupando las tareas según el tipo de actividad que sean. Además, se establecieron tres niveles de prioridad a cada grupo, y se estableció un conjunto inicial de tareas a implementar:

- Grupo de **prioridad 1** (alta). Son aquellas que pertenecen a las ABVD. Tienen la prioridad máxima al ser imprescindibles para una vida independiente.
 - **Levantarse.** Es necesario controlar que la persona se levanta a la hora asignada en su rutina, ya que si se levanta más tarde rompería con el plan diario y, si ocurre muy a menudo, podría significar algún estado mental negativo como la depresión.
 - **Aseo personal.** Una buena higiene personal evita posibles enfermedades, además de que aporta bienestar a uno mismo.
 - **Desayuno.** Es la primera comida del día, es recomendable que la persona la haga para comenzar la actividad diaria.
 - **Comida mediodía.** Es otra de las comidas principales.
 - **Actividad de movilidad física.** Una persona de edad avanzada necesita hacer un poco de ejercicio para mantener su estado físico.
 - **Merienda.** Otra de las comidas que se deben hacer al día.

- **Cena.** Última comida antes de irse a dormir.
- **Acostarse.** Es necesario controlar que se acuesta a ciertas horas para cumplir con los periodos de sueño establecidos.
- Grupo de **prioridad 2** (media). Son las que se incluyen en las AIVD.
 - **Limpieza básica de la casa.** Tener la casa limpia es un factor positivo para la salud mental y física del sujeto. Requiere un esfuerzo físico superior respecto a otras tareas. Tanto puede ser sólo pasarle la aspiradora a la casa y hacer la cama como limpiar los baños; esto dependerá del usuario.
 - **Salir a hacer la compra.** Esta tarea aunque, en la teoría, sería de prioridad 2, en la práctica también se le podría considerar de prioridad 1, ya que a la experta le parecía muy importante que las personas mayores salieran por lo menos una vez al exterior cada día. En este caso también se haría la compra de la comida de mediodía, y se evitaría el riesgo de que no haya suficientes alimentos a la hora de comer.
 - **Hacer la comida.** Es necesario preparar la comida para su posterior consumición, aunque pueda comer sin tener que prepararla. El beneficio que aporta esta tarea es que mantiene al usuario ocupado y estimula sus capacidades cognitivas y motrices.
 - **Limpiar la cocina.** Como limpieza básica pero limitándose al área de la cocina. Suele ser después de comer, ya que es cuando la cocina queda más sucia.
 - **Juego cognitivo.** Se proponen diferentes juegos que estimulen la capacidad cognitiva de la persona.
 - **Actividad de reminiscencia.** Se estimulará para que recuerde eventos de su vida pasada. Este tipo de actividades les suele gustar mucho a las personas mayores, ya que les gusta recordar, y es una buena manera de mantener su salud mental.
 - **Llamar a un familiar o amigo/a.** Para promover la comunicación social de la persona, es bueno que hable con alguien todos los días. De esta manera se puede disminuir el sentimiento de soledad que pueda sentir.
 - **Tomar la medicación.** Es importante que la persona mantenga adecuadamente su salud, por eso viene bien recordarle de que se tome la medicación asignada por su médico.
- Grupo de **prioridad 3** (baja). Son las tareas relacionadas con las AAVD. Son las menos prioritarias.
 - **Descanso y/o ocio.** Pueden ser actividades como ver la televisión, escuchar música, echar la siesta, leer un libro, entre otras. Es una tarea necesaria del horario ya que es importante mantener algunos periodos de descanso entre las actividades.

- **Salir al exterior.** Comprende todo aquello relacionado a actividades de ocio en el exterior. Esto es, ir a asociaciones de vecinos a clases de pintura (por ejemplo), pasear, visitar a familiares o amigos, etc.

Aunque se hayan establecido estas prioridades previamente, no serán asignadas por defecto en la implementación de la tarea, sino que serán configurables. Esto es porque el médico de cabecera del usuario (u otra persona que se haga cargo de él) puede considerar que ciertas tareas son más importantes que otras. La clasificación realizada se utilizará a la hora de escribir una primera aproximación del que sería, a modo de ejemplo, un horario estándar en el fichero de configuración.

Tarea genérica

Con el fin de facilitar la inclusión de nuevas tareas en el sistema, se diseñó una tarea genérica que establece la plantilla que seguirán todas las tareas en su implementación.

Una tarea consta de cuatro atributos:

- Su **prioridad**, que ya se explicó anteriormente.
- Su **duración**. Es el tiempo máximo, en minutos, disponible para la tarea en el peor de los casos; es decir, lo que tardaría la tarea en realizarse si hubo que animar al usuario para que la hiciera con todos los intentos disponibles.
- El **conjunto de estados de ánimo**, contiene todos los posibles estados de ánimo en los que se tiene que encontrar el usuario para poder realizar la tarea (los estados permitidos). Esto es debido a que puede haber tareas que necesiten que la persona esté de buen humor para poder hacerla. Los valores que puede tomar son los mismos que en la detección del estado de ánimo: Muy desanimado, Desanimado, Normal, Animado, Muy animado.
- La variable booleana **anormal**. Indica si su realización hace uso de la detección del estado de ánimo (*False*) o no (*True*). Se le denomina como anormal porque SARDAM, en general, se basa en ejecutar ciertas acciones dependiendo del estado de ánimo del usuario, con el fin de favorecer su cumplimiento.

En la Figura 5.5 se muestra el pseudocódigo de dicha plantilla. Primero se define la variable *tarea_fin*, que se utilizará para saber si la tarea ha finalizado o no. A continuación se define el número máximo de intentos que habrá para animar al usuario para que la haga, en caso de que no se encuentre apto de ánimo. El valor dependerá de la prioridad que esté asignada a la tarea; si es de prioridad 1 dispondrá de tres intentos, si es de prioridad 2 serán dos intentos, y si es la más baja (3) habrá un solo intento. Después se calcula la hora a la que finaliza la tarea,

INICIO

```

1. tarea_fin = False
2. intentos = max_intentos # valor por defecto 3 si prioridad es 1, 2 si prioridad es 2, y
   1 si prioridad es 3
3. hora_fin <= Ejecutar calcular_hora_final(duracion)
4. Si anormal == True entonces
    Ejecutar acciones_tarea( hora_fin)
    estado_fin <= ejecutar detectar_animo()
    Ejecutar guardar_log(True, estado)
5. Si no entonces
    Ejecutar proponer( )
    Mientras tarea_fin == False
        estado <= ejecutar detectar_animo( )
        Si estado == estados_permitidos entonces
            Ejecutar acciones_tarea( hora_fin)
            Ejecutar guardar_log(True, estado)
            tarea_fin = True
        Si no entonces
            Si intentos > 0 entonces
                Si intentos == max_intentos entonces
                    Ejecutar convencer( )
                Si no entonces
                    Ejecutar tarea_correctora ( )
                intentos = intentos - 1
            Si no entonces
                Si prioridad == 3 entonces
                    Ejecutar acciones_tarea( hora_fin)
                    Ejecutar guardar_log(True, estado)
                Si no entonces
                    Ejecutar guardar_log(False, estado)
                    Ejecutar proponer_descanso(hora_fin)
                tarea_fin = True
            Fin Si
        Fin Si
    Fin Mientras
Fin Si

```

FIN

Figura 5.5: Pseudocódigo de la plantilla de una tarea.

sumando su duración a la hora actual. En el siguiente paso, el 4, se comprueba si se trata de una tarea anormal; y si es el caso se ejecutan todas las acciones que tiene que realizar hasta finalizar (*acciones_tarea(hora_fin)*) y después se registra que sí la hizo, guardando también su estado de ánimo al finalizar para mantener un seguimiento de su estado emocional. Si es una tarea normal (paso 5), se empieza proponiéndosela al usuario. Se detecta el estado de ánimo; si se encuentra en los estados permitidos entonces se ejecutan las acciones de la tarea hasta su finalización. Si no está del ánimo esperado, se intenta convencer por primera vez. Si después de tratar de convencerlo sigue negándose a realizar la tarea, se le propone una tarea correctora en caso de que queden más intentos. Estas tareas son una de las actividades de duración mínima definidas para levantar el ánimo a la persona. Si el usuario finalmente accede a realizar la tarea, se ejecuta el resto de la tarea (las acciones). En cambio, si no se consiguió mejorar su estado de ánimo, la tarea le propone descanso, excepto si es de prioridad 3. En ese caso, como las tareas de prioridad 3 no se controlan al no ser determinantes para la vida independiente de la persona, se ejecutan las acciones de la tarea igual. En cualquiera de los casos queda registrado si la tarea se realizó o no y el estado de ánimo.

La Figura 5.6 muestra el diagrama de flujo que indica cómo se comporta una tarea en base al estado de ánimo del usuario.

5.6.2 Diseño e implementación de SARDAM

A partir del conjunto de tareas definido, se desarrolla el sistema de monitorización. Es el componente principal, ya que es el que hace la función principal de SARDAM: ejecutar y controlar cada tarea de la rutina diaria.

```
[Inicio]
inicio = 8:45

[Levantarse]
prioridad = 1
duracion = 5
estados_de_animo =
anormal = True

[AseoPersonal]
prioridad = 1
duracion = 20
estados_de_animo = Muy animado-Animado-Normal
anormal = False

[Desayuno]
prioridad = 1
duracion = 30
estados_de_animo = Muy animado-Animado-Normal
anormal = False

[LimpiezaBasica]
prioridad = 2
duracion = 60
estados_de_animo = Muy animado-Animado
anormal = False
```

Figura 5.7: Ejemplo del fichero de configuración del horario.

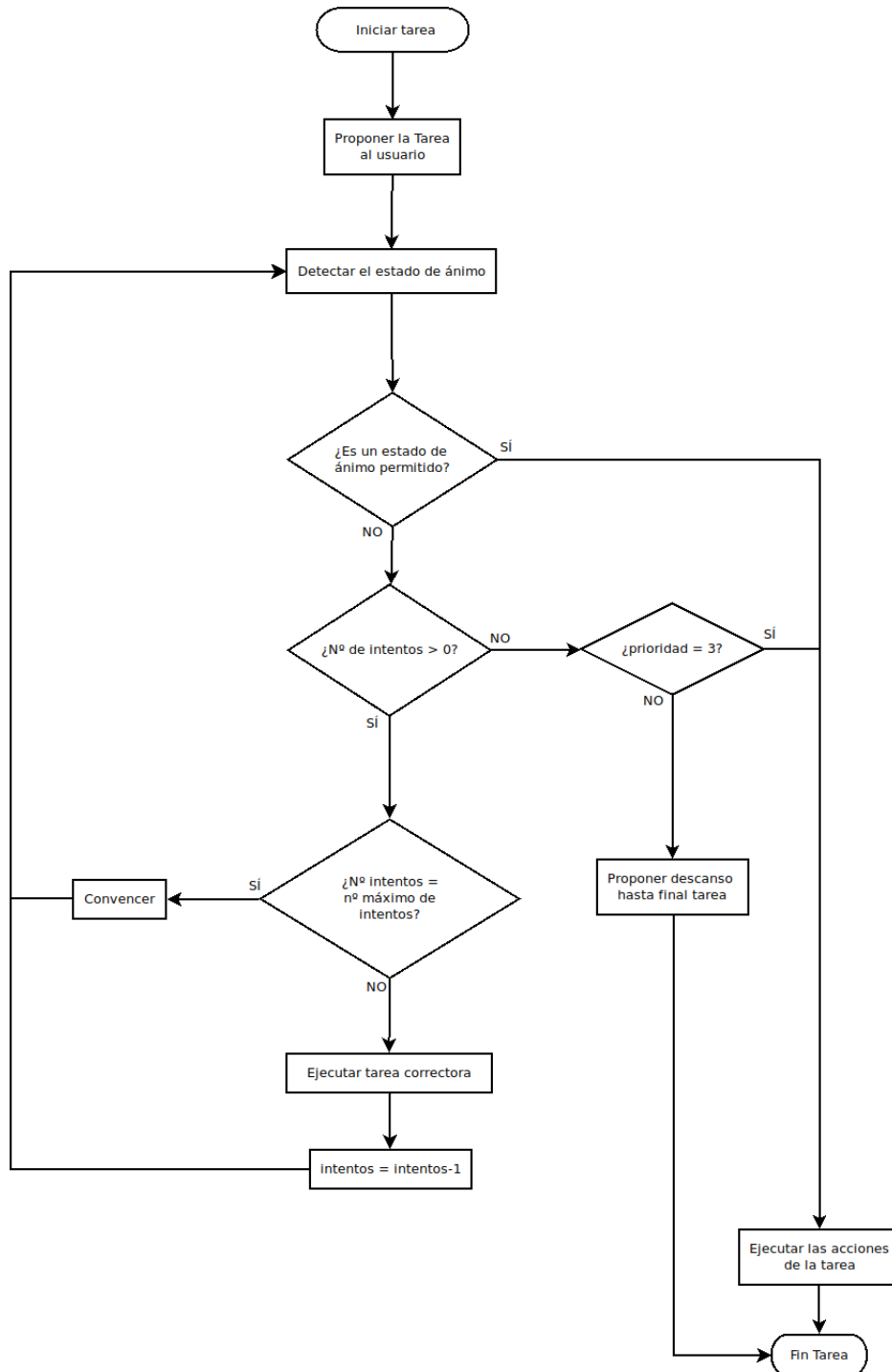


Figura 5.6: Diagrama de flujo tarea genérica.

El horario, con la fecha de inicio del día y el conjunto de tareas con sus atributos, es definido en un fichero de configuración (horarioX.cfg). Un ejemplo de este fichero de configuración se puede ver en la Figura 5.7. Respecto a dicho horario, si se quiere poner varias veces una misma tarea, hay que denominarlas como *nombretarea_X*, siendo X un número. El sistema permite seleccionar qué horario se quiere ejecutar de los definidos, ya que se pueden tener varios ficheros con las diferentes rutinas diarias. La selección sería ejecutando el fichero main.py con la opción “-horario” y a continuación el nombre del fichero sin la extensión .cfg, como se muestra en la Figura 5.9.

En la Figura 5.8 se puede ver los módulos (5.8a), y carpetas que componen a SARDAM (5.8b), incluyendo los ficheros de configuración (horarioX.cfg, nao.cfg) y el fichero log que mantiene el registro de las tareas (monitorización.log). La ejecución del sistema comienza en main.py, donde se procesa el horario del fichero de configuración para que sea apto para usar en Sistema. Como se puede ver, cada tarea se implementa en un módulo diferente. Así, podemos ver, por ejemplo, el módulo levantarse o juego_palabras. Este código se encuentra subido en el GitLab de la FIC ¹.

Respecto al diseño de SARDAM, la Figura 5.10 muestra el diagrama de clases resultante de SARDAM. Se hizo de manera que cumpliera con los requisitos establecidos, es decir, que fuera modular, mantenible y reutilizable. Permite añadir actividades y tareas fácilmente; simplemente se hereda de dichas clases abstractas y se siguen los pasos para implementar cada una de ellas. También se trató que cumpliera con todos los principios de diseño de programación orientada a objetos, haciendo uso de patrones de diseño si así lo requería. En dicho diagrama destacan cuatro clases: Sistema, Funcionalidades, Tarea y Actividades.

```
/SARDAM$ python main.py --horario horario_1
```

Figura 5.9: Ejecutar SARDAM.

Sistema

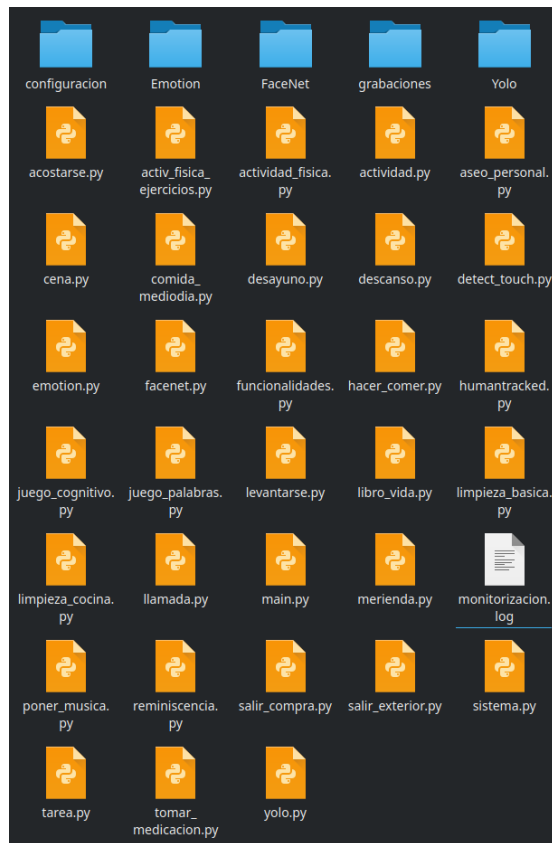
El objeto Sistema se encarga de gestionar la ejecución de las tareas del horario.

El constructor de esta clase tiene dos atributos: **horario**, que es la lista de tareas con su nombre, prioridad, duración, conjunto de estados de ánimo permitidos y si es anormal o no (*True* o *False*), y la **hora de inicio** del día, en formato HH:MM.

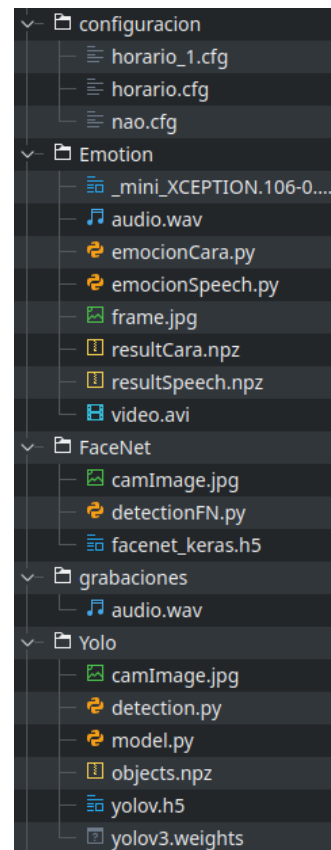
Desde su función *ejecutar_horario()*, se ejecuta cada tarea de la lista en el orden que se ha obtenido del horario hasta finalizar o la ejecución sea interrumpida por el usuario (por el teclado).

Este objeto es iniciado desde main.py, que es quien extrae el horario y la hora de inicio del fichero de configuración y los prepara para pasárselos al constructor del objeto Sistema.

¹<https://git.fic.udc.es/c.mlamas/SARDAM.git>



(a) Módulos de SARDAM.



(b) Contenido de las carpetas.

Figura 5.8: Lista de ficheros del sistema SARDAM.



Funcionalidades

En la clase Funcionalidades se encuentran los métodos que engloban las funciones que se utilizan del robot, los detectores de NAOqi que se han creado y los sistemas de detección y reconocimiento. De esta manera se encapsulan las distintas funcionalidades del sistema, evitando tener que repetir código y creando métodos que definan nuevas funcionalidades a mayores de las que ofrece el robot. Su diseño se basa en el patrón fachada.

En el constructor se crean todos los objetos de los módulos de NAOqi, accediendo a ellos a través de ALProxy, como se indicó en la Sección 3.1.1. La dirección IP y puerto del robot se obtiene a través del fichero de configuración nao.cfg. Además se crean los objetos del sistema de detección del estado de ánimo, YOLO y FaceNet.

Los métodos de esta clase se dividen en tres bloques. Los del primer bloque son los que están relacionados con las funcionalidades del robot son *habla(texto)*, *emitir_audio(fichero_audio)*, *grabar_audio(tiempo, carpeta)*, *grabar_camara(tiempo, carpeta)*, *grabar(tiempo, carpeta)*, *capturar_imagen(carpeta, camara)*. Con ellos se intenta desacoplar a SARDAM del robot utilizado, de manera que si en un futuro se cambia de robot sólo haya que rehacer estos métodos. A continuación se resume el objetivo de cada método mencionado:

- *habla(texto)*: Es el que se encarga de que el robot diga el texto que recibe como parámetro. Para ello se utiliza la función *say()* del módulo ALTextToSpeech.
- *emitir_audio(fichero_audio)*: Hace que el robot emita el audio cuyo nombre recibió como parámetro. Para ello dicho fichero tiene que encontrarse en el robot para que pueda ejecutarlo. Utiliza las funciones del módulo ALAudioPlayer.
- *grabar_audio(tiempo, carpeta)*: Graba con los micrófonos del robot durante el periodo de tiempo indicado, y guarda el fichero en la carpeta que se le indica. Para ello se utilizan los métodos del módulo ALAudioRecorder.
- *grabar_camara(tiempo, carpeta)*: Graba con la cámara superior del robot durante el periodo de tiempo indicado, y guarda el vídeo en la carpeta que se le indica. Se hace uso de las funciones de ALVideoRecorder.
- *grabar(tiempo, carpeta)*: Graba con la cámara superior y con los micrófonos a la vez durante el periodo de tiempo indicado, y guarda los ficheros de audio y vídeo en la carpeta que se le indica por parámetro. Aquí se utilizan los métodos de los módulos ALAudioRecorder y ALVideoRecorder.
- *capturar_imagen(carpeta, camara)*: Captura una imagen desde la cámara del robot y la guarda en la carpeta indicada. Con la variable *camara* es número entero que indica si se quiere que se capture con la cámara superior (0) o con la inferior (1).

El otro bloque de métodos son los dos que ejecutan los módulos creados de NAOqi para la detección de personas (*hay_personas()*) y la detección del contacto del usuario con el robot (*detectar_tacto()*).

En el último bloque se encuentran los métodos que ejecutan el sistema de detección del estado de ánimo, FaceNet, YOLO y Google SpeechRecognition:

- *detectar_animo()*: Graba durante 5 segundos con los micrófonos del NAO, y a continuación ejecuta el detector del estado anímico.
- *reconocer_persona()*: Primero captura una imagen del robot y se guarda en la carpeta de FaceNet. Si ya existen personas en la base de datos, se ejecuta la función *detectar()* de la clase FaceNet para reconocer a la persona. Si no o no sabe quién es después de detectar FaceNet, entonces procede a guardarla en la base de datos: primero le pregunta por su nombre, y después captura 10 imágenes para la base de datos de entrenamiento, y 5 imágenes para la base de datos test. A continuación procede a reentrenar el modelo (obteniendo los nuevos *face embeddings*) Utiliza los módulos ALVideoDevice, ALSpeechRecognition, ALTextToSpeech y ALMemory. En este proyecto no se llegó a utilizar esta función.
- *detectar_objetos()*: Captura la imagen de la que se quiere detectar con la cámara inferior del robot, y a continuación se ejecuta el método de la clase Yolo que detecta qué objetos hay en la imagen.
- *reconocer_habla_audio(fichero_audio)*: Se detectan las palabras que se encuentren en el fichero de audio recibido como parámetro.

Tarea

En esta clase se aplica el patrón plantilla, ya que la lógica de una tarea es prácticamente igual, solo cambia las acciones que ejecuta la tarea, como se vio en el apartado anterior con la tarea genérica. Esta clase abstracta llamada *Tarea* contiene todos los métodos que deben implementar las tareas que se desarrollen, además del método principal, que es la plantilla diseñada en la Sección 5.6.1. Además, también define las cuatro variables abstractas que las clases hijas inicializarán en el constructor, y el objeto Funcionalidades.

Para crear el objeto se necesita pasarle al constructor la **prioridad**, la **duración**, la lista con los **estados de ánimo** permitidos y el **booleano** que indica si es una tarea anormal o no.

Como se puede ver en el diagrama de clases (Figura 5.10), los métodos de la clase Tarea son:

- *realizar_tarea()*: Esta función es la que implementa la tarea genérica mencionada previamente.

- *proponer()*: Este es uno de los métodos abstractos que las tareas específicas (las clases que heredan de ésta) deben implementar. En este método solamente se le propone la tarea al usuario.
- *convencer()*: Este es otro de los métodos abstractos. En él el robot debe de intentar convencer al usuario con alguna frase motivadora.
- *acciones_tarea(hora_fin)*: Este es el último de los métodos abstractos que se deben implementar. Tiene que contener las acciones específicas de la tarea.
- *tarea_correctora()*: En este método se ejecuta una de las actividades (ActivFisicaEjercicios, JuegoPalabras, PonerMusica, LibroDeVida) para levantar el ánimo del usuario. Dependiendo de la duración de la tarea, esta actividad correctora puede durar 5 minutos (si duración es menor o igual a 30 minutos) o 10 minutos (si duración mayor a 30 minutos) aproximadamente.
- *proponer_descanso(hora_fin)*: Este método se implementa en esta clase, ya que es común para todas las tareas. Es el que le propone al usuario tomarse un descanso hasta que finalice el tiempo de la tarea en caso de que no se llegue a ejecutar.
- *guardar_log(hizo_tarea, estado)*: Esta función también se implementa ya en la clase Tarea. Se encarga de guardar en el registro (log) si hizo la tarea y cuál era su estado de ánimo.
- *guardar_info(info)*: Esta función, que también se implementa en Tarea, se encarga de guardar la información que se desee en el registro.

Las tareas que se desarrollaron para esta primera versión del sistema son las listadas en la Sección 5.6.1. A continuación se describen cuatro de estas tareas, indicando cuál son los sensores y actuadores que utilizan, y las acciones que realizan.

Tarea **Levantarse**:

- Prioridad: 1
- Sensores: HumanTracked.
- Actuadores: ALTextToSpeech, ALAudioPlayer.
- Acciones:
 1. El robot emite el audio del despertador.
 2. Cuando termina el audio, comprueba si hay alguien frente a él (con la función *hay_personas()*).

3. Si encuentra a el usuario, entonces el robot le dice “Buenos días” y se ejecuta el paso 9.
4. Si no, entonces se vuelve a emitir el sonido del despertador.
5. El robot vuelve a comprobar si el usuario está frente a él.
6. Si no hay nadie se espera 30 segundos.
7. Si aún no ha finalizado el tiempo de la tarea, se vuelve otra vez al paso 2.
8. Si el tiempo finalizó, se ejecuta el paso siguiente.
9. La tarea termina.

Tarea **AseoPersonal:**

- Prioridad: 1
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para asearse.
 2. Se registra en el fichero de monitorización la hora en ese momento.
 3. Se esperan 20 segundos para dar tiempo al usuario entrar en el baño y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar aseándose.
 4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
 5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **Desayuno:**

- Prioridad: 1
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para desayunar.

2. Se registra en el fichero de monitorización la hora en ese momento.
3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea ActividadFisica:

- Prioridad: 1
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder, ALMotion, ALRobotPosture.
- Acciones:
 1. Se calcula la duración que tendrá la actividad. Esto se hace calculando el tiempo que queda desde ese momento a la hora final, que será lo que dure la actividad.
 2. El robot contesta diciendo “Muy bien. Vamos a hacer ejercicios de movilidad física”.
 3. Se ejecuta la actividad ActivFisicaEjercicios con la duración estimada.
 4. Una vez finalizada la actividad, la tarea termina.

En el Apéndice B se describen el resto de las tareas implementadas.

Actividad

Actividad es la clase abstracta de la que heredan todas las Actividades que se implementen en el sistema. Tiene la variable abstracta duración, la cual deben incluir las clases hijas, recibéndola a través del constructor, y el objeto Funcionalidades. Tiene un único método abstracto que las clases hijas deben de implementar: *ejecutar_actividad()*.

La diferencia de una Actividad a una Tarea es que la Actividad es una clase que contiene un conjunto de acciones con un objetivo común que puede tanto formar parte de la tarea como ser usada como tarea correctora. Están diseñadas para poder ser reutilizadas, ya que pueden ser usadas en más de una tarea, con la duración que se desee (se pueden ejecutar para poco tiempo como para mucho más) e incluso pudiendo usar más de una en dicha tarea. Una

Actividad también es una tarea correctora, ya que su función se basa en el entretenimiento del usuario y están lo suficientemente capacitadas para levantar el ánimo al usuario.

Como se puede observar en el diagrama de clases mencionado previamente (Figura 5.10), hay cuatro clases que heredan de ésta; son **PonerMusica**, **ActivFisicaEjercicios**, **JuegoPalabras** y **LibroDeVida**. Éstas son las cuatro actividades que se desarrollaron para esta primera versión de SARDAM.

La primera, **PonerMusica**, ejecuta canciones (en este caso, dos ficheros de audio), que se encuentran en el robot, durante el intervalo de tiempo establecido, aproximadamente. No corta la canción por la mitad cuando se acaba el tiempo, sino espera a que termine para finalizar la actividad. Las canciones utilizadas en este proyecto fueron descargadas de la página web Free Music Archive [35].

En **ActivFisicaEjercicios** se define una rutina de ejercicios, donde sus repeticiones van a depender de la duración de la actividad (para no cortar la rutina por la mitad):

- Si dura 5 minutos, entonces se repite cada ejercicio 5 veces.
- Si dura 10 minutos, serán 10 repeticiones.
- Si son 15 minutos, las repeticiones serán 15.
- En cambio, si son 20 minutos, se repetirá la rutina dos veces, con 10 repeticiones en los ejercicios.

Cabe destacar que la rutina, con 5 repeticiones por cada ejercicio, dura sobre unos 6 minutos. Se espera que la rutina no dure mucho más de los 20 minutos, ya que son pocos ejercicios y para tiempos superiores se harían muy repetitivos y cansinos. Cabe destacar que la duración será aproximada a la establecida, ya que cada ejercicio tiene sus tiempos.

La rutina está compuesta por siete ejercicios, como se puede ver en la Figura 5.11. Estos ejercicios se obtuvieron de fichas y vídeos sobre gerontogimnasia. El primer ejercicio (Figura 5.11a) consiste en, con los brazos paralelos al suelo y formando un ángulo de noventa grados, abrir y cerrar los brazos. En el segundo de los ejercicios (Figura 5.11b) se mueven la muñecas hacia fuera y hacia adentro. En el tercer ejercicio (Figura 5.11c) hay que mover la cadera dibujando círculos con ella, primero hacia un lado y después hacia el otro. El cuarto ejercicio (Figura 5.11d) es como el anterior sólo que con la cabeza; se mueve hacia un lado y después hacia el otro. Se indica al usuario que coja las mancuernas o otro objeto de peso. En el quinto ejercicio (Figura 5.11e), ya con las mancuernas y con los brazos paralelos al suelo y en ángulo de noventa grados, se estiran y se vuelven a contraer hacia la cara. El sexto de los ejercicios (Figura 5.11f) consiste en levantar los brazos a la vez y a continuación bajarlos. Por último, en el ejercicio 7 (Figura 5.11g), se levanta primero un brazo y después es el otro, como si se estuviera saludando.

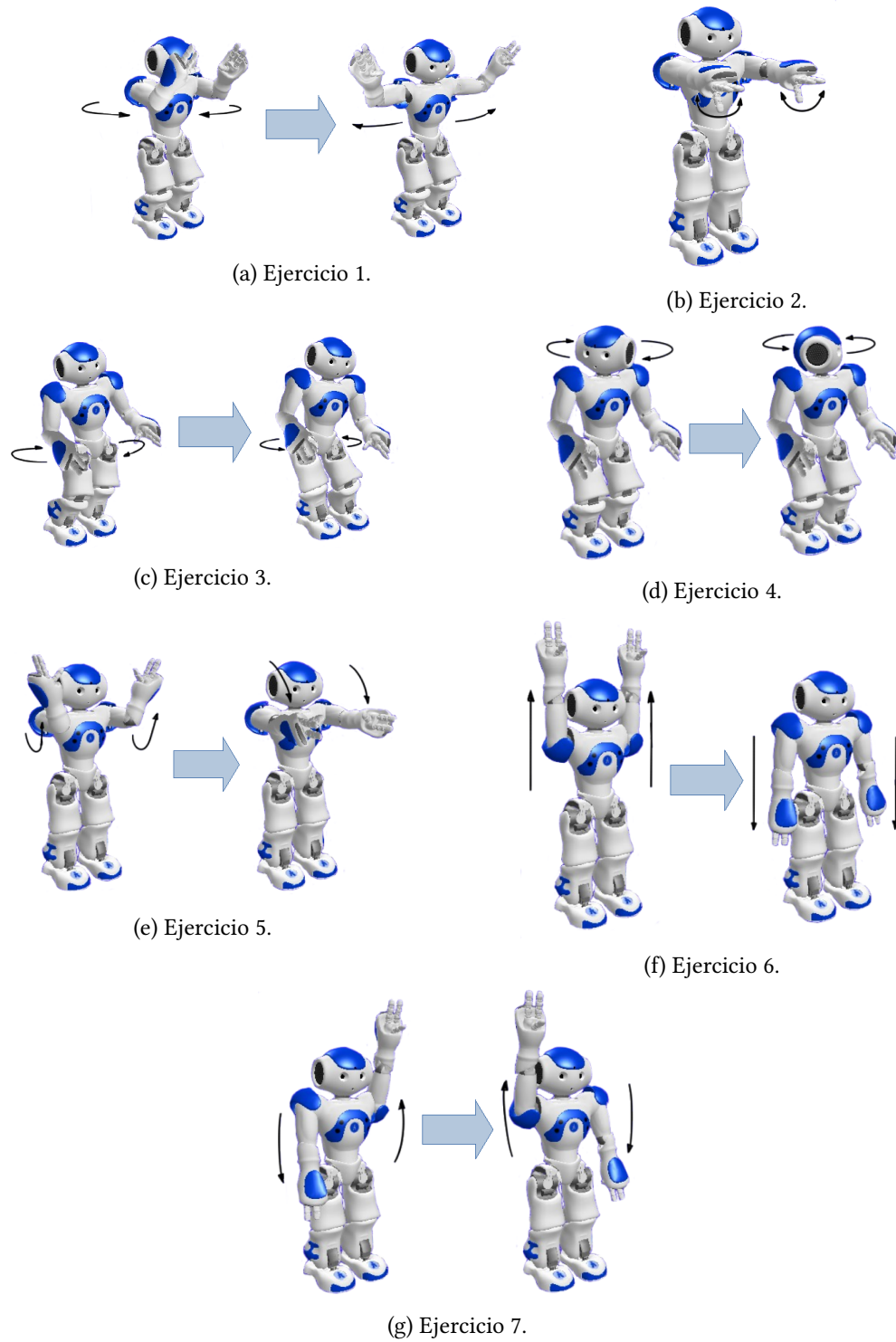


Figura 5.11: Rutina de ejercicios de movilidad física.

LibroDeVida implementa una actividad de reminiscencia basándose en la comentada por la psicóloga en la segunda captura de requisitos (Sección 5.1.2), también llamada "Libro de vida". Se hizo una versión básica limitada a las posibilidades que ofrecían las funcionalidades de SARDAM y del robot, de manera que sirviera de ejemplo a cómo serían este tipo de actividades de reminiscencia. Hay que tener en cuenta que la detección de objetos reconoce objetos generales, que pueden ser desde unas tijeras a un coche. Considerando que el usuario ya recibió instrucciones previamente para la realización del libro, al inicio de la actividad el robot le propone que coja uno o dos objetos que le recuerden a eventos de su vida pasada y los coloque frente a él. Cuando los tenga, tiene que tocarle alguna de las manos o la cabeza para indicarle al robot que ya ha obtenido los objetos y los ha puesto frente a él. NAO detecta cuáles son los objetos, y, dependiendo de la duración de la actividad, le pide que escriba o represente en el cuaderno sobre uno de los objetos (si duración menor o igual a 15 minutos) o sobre los dos (repartiendo el tiempo a la mitad por cada elemento). Lo que se pretende es que se rellene el cuaderno con recuerdos, donde se puede dibujar y añadir fotos, y así en un futuro, si empieza a desarrollar un deterioro cognitivo más notable, pueda seguir estimulando el cerebro leyendo del libro de vida. En la Figura 5.12 se puede ver un ejemplo gráfico de cómo sería la ejecución de esta actividad. Como se puede observar deja 10 minutos para cada objeto de los 20 minutos que tiene asignados.

La última actividad implementada, **JuegoPalabras**, es un juego cognitivo en el que el robot le dicta una lista de elementos sobre una temática, y posteriormente el usuario tiene 20 segundos para contestar con todos los que recuerde. En este caso son tres temas: fruta, animales y ciudades de España.

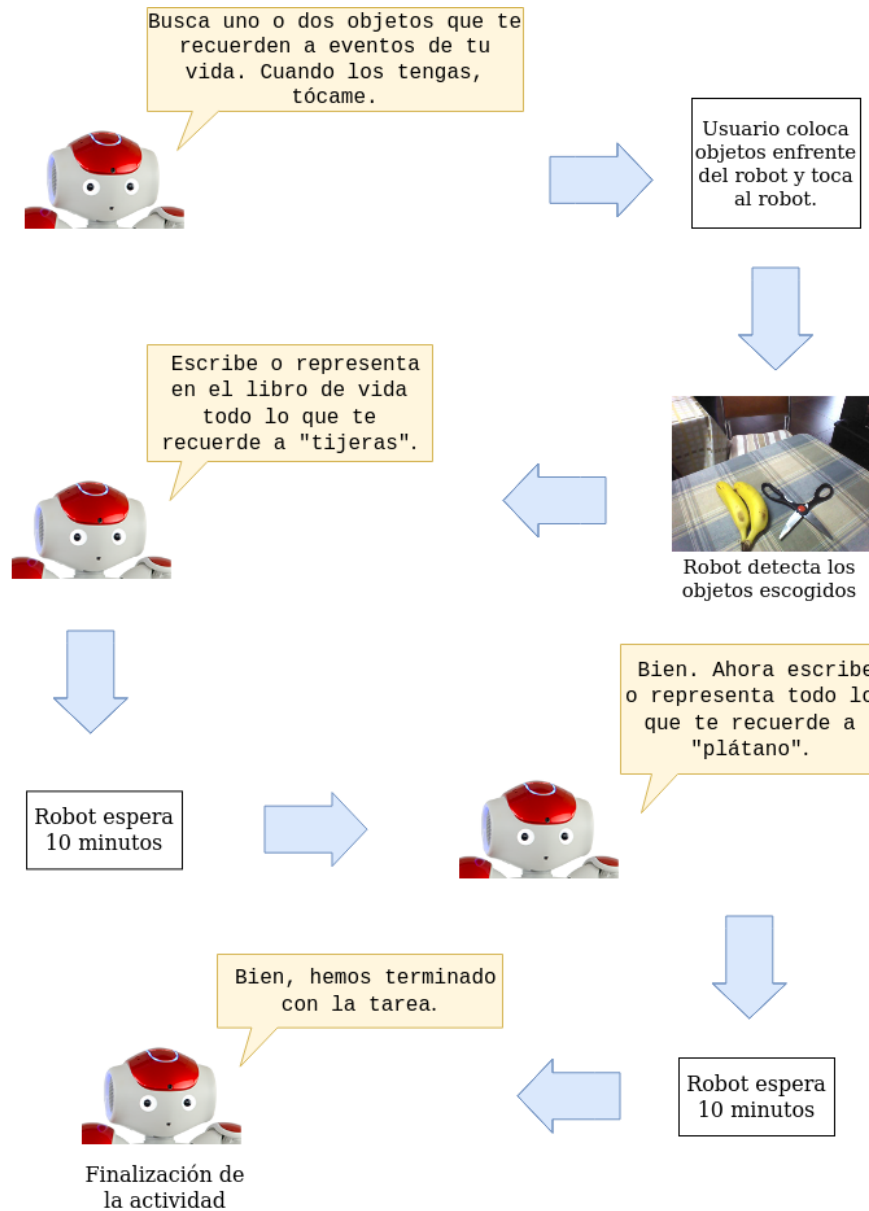


Figura 5.12: Ejemplo gráfico de ejecución de la actividad LibroDeVida con duración de 20 minutos.

Capítulo 6

Pruebas

PARA probar el funcionamiento de SARDAM, se realizó una prueba durante un día entero con una mujer de 65 años, que llamaremos María para preservar su identidad, sobre un escenario predefinido.

6.1 Antes de empezar

Se establece la rutina diaria del usuario en el fichero de configuración para el horario, que se ajusta a sus preferencias:

- Hora inicio día: 9:00h.
- Levantarse.
 - Prioridad: 1
 - Duración: 5 minutos
 - Estados de ánimo permitidos: Ninguno (es una tarea anormal)
 - Anormal: *True*
- Desayuno.
 - Prioridad: 1
 - Duración: 30 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Aseo personal.
 - Prioridad: 2

- Duración: 20 minutos
- Estados de ánimo permitidos: Muy animado, Animado, Normal
- Anormal: *False*
- Limpieza básica de la casa.
 - Prioridad: 2
 - Duración: 1 hora y 20 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Salir a la compra o exterior.
 - Prioridad: 1
 - Duración: 60 minutos
 - Estados de ánimo permitidos: Muy animado, Animado
 - Anormal: *False*
- Hacer de comer.
 - Prioridad: 2
 - Duración: 1 hora y 30 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Comida mediodía.
 - Prioridad: 1
 - Duración: 45 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Limpieza de la cocina y otras tareas del hogar.
 - Prioridad: 2
 - Duración: 45 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*

- Descanso 1.
 - Prioridad: 3
 - Duración: 1 hora y 25 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Actividad de movilidad física.
 - Prioridad: 1
 - Duración: 20 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Juego cognitivo.
 - Prioridad: 2
 - Duración: 15 minutos
 - Estados de ánimo permitidos: Muy animado, Animado
 - Anormal: *False*
- Merienda.
 - Prioridad: 1
 - Duración: 15 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Actividades de ocio en el exterior.
 - Prioridad: 3
 - Duración: 1 hora y 15 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Actividad de reminiscencia.
 - Prioridad: 2
 - Duración: 30 minutos

- Estados de ánimo permitidos: Muy animado, Animado
 - Anormal: *False*
- Llamada a un familiar o amigo.
 - Prioridad: 2
 - Duración: 30 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Cena.
 - Prioridad: 1
 - Duración: 45 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Tomar la medicación.
 - Prioridad: 1
 - Duración: 15 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Descanso 2.
 - Prioridad: 3
 - Duración: 45 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*
- Acostarse.
 - Prioridad: 2
 - Duración: 15 minutos
 - Estados de ánimo permitidos: Muy animado, Animado, Normal
 - Anormal: *False*

Se predefine una personalidad para este escenario, en base a la experiencia de los expertos, que representa uno de los posibles casos con los que el sistema puede encontrar en una situación real. De esta manera, se puede ver cómo reaccionaría el sistema a este tipo de situaciones. En este caso, la usuaria va a ser una persona que no le gustan mucho las tareas de mucha movilidad como Limpieza básica y Actividad física. Por las tardes está de peor humor que por las mañanas; se suele quejar cuando es hora de merendar, llamar a alguien o cuando hay que irse a dormir. Por lo tanto, se establece que las tareas en las que no va a estar de humor son:

- En LimpiezaBasica se quejará dos veces en las que el robot le pregunte.
- En HacerComer también se quejará dos veces.
- En ActividadFisica estará de mal humor la primera vez cuando le propone la tarea.
- No querrá hacer Merienda una vez, al proponerle la tarea.
- En la tarea Llamada se quejará todas las veces que el robot le pregunte qué tal está.
- En Acostarse no querrá hacer la tarea cuatro veces.

El robot se posiciona de manera que se vea la puerta de la entrada y un poco en dirección hacia el “pasillo”, como se ve en la Figura 6.1. Se colocó de esta manera porque las tareas sobre las comidas controlan cuando vuelven a ver a la usuaria, por lo que así no se puede ver la cocina. Esta situación es la ideal en SARDAM, ya que el robot puede gestionar todo el espacio.

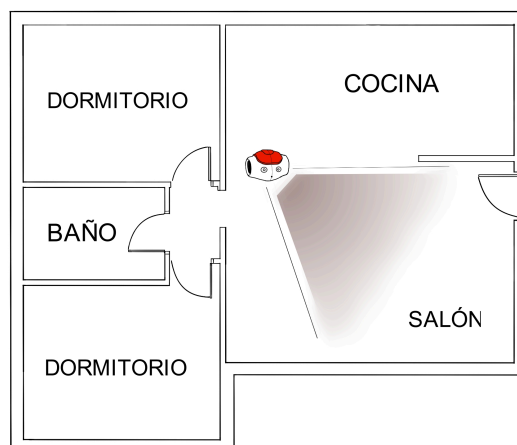


Figura 6.1: Colocación del robot en la casa, con su campo de visión.

6.2 Ejecución de las pruebas

Se ejecutó el fichero main.py para iniciar SARDAM, empezando la rutina a las 9:00h. Durante la ejecución, las tareas que se ejecutaron con normalidad fueron: Levantarse (Figura 6.2), Desayuno, AseoPersonal, SalirCompra, ComidaMediodia, LimpiezaCocina, DescansoOcio_1, JuegoCognitivo, SalirExterior, Reminiscencia, Cena, TomarMedicacion, DescansoOcio_2.

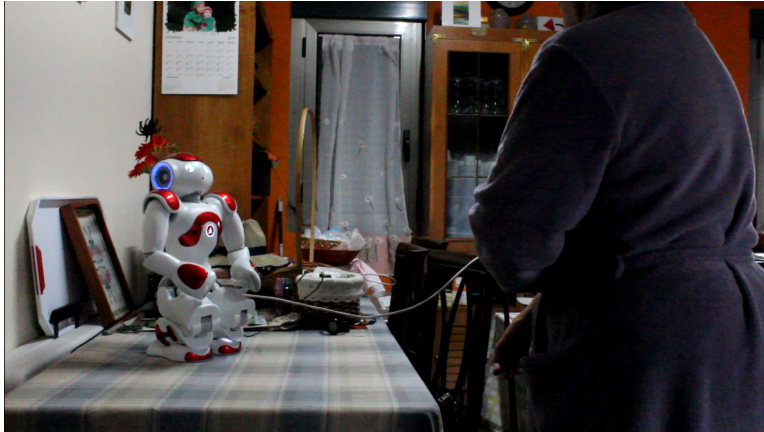


Figura 6.2: Prueba de la tarea Levantarse.

Para el resto de tareas, se explica más detalladamente cada una a continuación. En LimpiezaBasica la usuaria estaba desanimada cuando el robot se la propuso, por lo que le intenta convencer. Como seguía de mal humor, el robot ejecutó una tarea correctora, que era JuegoPalabra, que sí que consiguió motivar a la usuaria.

A la hora de la ActividadFisica, la usuaria se quejó una vez y el robot le intentó convencer. Después de convencer, la usuaria se encontraba de buen humor pero el robot detectó que seguía de mal humor (en detección por cara dio desanimado), por lo que ejecuta una tarea correctora (JuegoPalabras). Después de esta tarea correctora ya detecta que la usuaria está bien de ánimo y se procedió con la actividad (Figura 6.3).

En Merienda la usuaria sí que se quejó una vez, donde el robot la intentó convencer. Pero, a continuación, cuando la usuaria ya estaba de buen humor, detectó que no estaba de humor cuando sí lo estaba (como en ActividadFisica).

Para la tarea Llamada la usuaria se quejó en todos los casos: en convencer y en las tareas correctoras propuestas, por lo que finalmente el robot le propuso descansar hasta la siguiente tarea.

Por último está Acostarse, donde la usuaria se queja tres veces: una cuando se le propone la tarea, otra después de convencer, y las otras después de realizar una de las dos tareas correctoras, donde se le propuso PonerMusica y Reminiscencia. Al finalizar la tarea de Reminiscencia, la usuaria accede a irse a dormir. Con esto el horario finaliza, terminando el día.



Figura 6.3: Ejercicio de movilidad física.

La Figura 6.4 muestra cómo quedó el fichero log que registra la realización de las tareas, monitorizacion.log.

```

Sí hizo Levantarse. Su estado de ánimo era Animado
Información sobre Desayuno: Usuario va a desayunar a las: 9:9
Información sobre AseoPersonal: Usuario va a asearse a las: 9:29
Sí hizo LimpiezaBasica. Su estado de ánimo era Animado
Información sobre SalirCompra: Usuario sale de casa a las: 11:5
Información sobre SalirCompra: Usuario visto a las: 11:41
Sí hizo SalirCompra. Su estado de ánimo era Animado
Sí hizo HacerComer. Su estado de ánimo era Animado
Información sobre ComidaMediodia: Usuario va a comer a las: 13:32
Información sobre ComidaMediodia: Usuario visto a las: 13:58
Sí hizo ComidaMediodia. Su estado de ánimo era Animado
Sí hizo LimpiezaCocina. Su estado de ánimo era Animado
Sí hizo DescansoOcio. Su estado de ánimo era Animado
Sí hizo ActividadFisica. Su estado de ánimo era Animado
Sí hizo JuegoCognitivo. Su estado de ánimo era Animado
Información sobre Merienda: Usuario va a merendar a las: 17:1
Sí hizo Merienda. Su estado de ánimo era Animado
Sí hizo SalirExterior. Su estado de ánimo era Animado
Sí hizo Reminiscencia. Su estado de ánimo era Animado
Sí hizo Reminiscencia. Su estado de ánimo era Animado
NO hizo Llamada. Su estado de ánimo era Desanimado
Información sobre Cena: Usuario va a cenar a las: 19:32
Información sobre Cena: Usuario visto a las: 20:9
Sí hizo Cena. Su estado de ánimo era Animado
Sí hizo TomarMedicacion. Su estado de ánimo era Animado
Sí hizo Descanso. Su estado de ánimo era Animado
Información sobre Acostarse: Usuario se fue a dormir a las: 21:6
Sí hizo Acostarse. Su estado de ánimo era Animado

```

Figura 6.4: Fichero monitorizacion.log al finalizar el día.

6.2.1 Errores y problemas durante la ejecución

La ejecución en términos generales fue bien, aunque hubo algunos problemas. Tanto en ActividadFisica como en Merienda no se detectó bien el estado de ánimo después del primer convencer, detectando un mal estado de ánimo y, consecuentemente, proponiendo la tarea correctora. Esto puede ser porque la usuaria es una persona poco expresiva, con emociones

planas, por lo que en esas veces el sistema de detección del estado de ánimo recurría al detector por audio.

En términos generales, unos de los problemas que se pueden encontrar en esta versión de SARDAM son una detección del estado emocional lenta y que dicha detección pueda equivocarse y no dar el estado de ánimo correcto (que sería un problema ya que obligaría al usuario a hacer una tarea correctora, incluso). Otro problema que se ha encontrado, en JuegoPalabras (además de que se haga muy repetitivo), muchas veces no detecta bien todas las palabras que le dice el usuario, especialmente cuando se trata de ciudades de España. También en la actividad PonerMusica se encontró otro problema (pero ya relativamente conocido) que es que su duración se puede extender más de la establecida al no terminar la actividad hasta que se termine la canción, siendo un problema cuando las canciones no llegan sobre los cinco o diez minutos establecidos. En la detección de objetos también se identificó un problema, y es que a veces no detecta bien los objetos o no los detecta todos.

6.3 Opinión del usuario

Después de finalizar la prueba, se le pregunta a la usuario que le ha parecido la experiencia. Comentó que ni le agobió ni le pareció invasivo a la hora de proponer y controlar las tareas. Sintió que el robot le hacía compañía, que podía cubrir esa parte emocional de sentirse menos sola al ver que algo le hablaba.



Figura 6.5: Robot y usuaria interactuando entre ellos.

Le pareció muy interesante lo de la detección emociones e intentar animarla para que hiciera las tareas. En especial destacó las tareas correctoras. Sentía que le hacían pensar y estar atenta a otras cosas y que la podían motivar para hacer la tarea posteriormente.

Además, consideraba que conectaba más con el robot que si fuera con otros dispositivos

como un ordenador o *tablet*, ya que le pareció menos frío en interacción que estos otros dispositivos. Sentía más empatía por él, como si fuese alguien que se preocupase por ella. Es más, se le preguntó si, en caso de que no hubiese el sistema de detección de emociones, la experiencia sería la misma para ella, y ella dijo que no, que prefería este sistema, ya que le daba esa sensación de que el robot tenía sentimientos.

En la Figura 6.5 se puede ver a la usuaria y al robot interactuando.

Conclusiones

EN este trabajo se consiguió desarrollar un prototipo de un sistema SAR enfocado en la atención de una persona de edad avanzada. SARDAM, que es así cómo se le denominó, se encarga de monitorizar la rutina diaria del usuario, motivándolo a que haga sus actividades básicas en caso de que la persona no se encuentre bien de ánimo para ello.

Se consiguió cumplir con todos los objetivos expuestos en la Sección 1.2. El primer objetivo, que era la capacidad de usar un robot humanoide y sus sensores (especialmente cámara, audio y movimiento), se logró en la primera iteración del proyecto, en la que se utilizó el robot NAO y su API NAOqi para probar sus distintas funcionalidades.

El segundo de los objetivos era utilizar algoritmos para el reconocimiento de personas con el NAO, en el que se usó el modelo FaceNet, y la detección de objetos, donde se utilizó YOLO.

Para lograr el tercer objetivo, que era utilizar algoritmos de reconocimiento de habla con el NAO, se usó el reconocimiento de palabras de NAOqi (ALSpeechRecognition) y la librería de Python SpeechRecognition, utilizando el reconocimiento de Google.

El cuarto objetivo se trataba sobre el uso de algoritmos de reconocimiento de emociones y con ellos desarrollar un sistema híbrido de detección del estado de ánimo para NAO. Para ello se utilizó la versión adaptada de FaceEmotion_ID, y se implementó un sistema de detección de emociones que clasifica las palabras encontradas en el audio en tres estados de ánimo (Desanimado, Normal, Animado). El sistema de detección final fusiona los resultados de dichos sistemas, siendo capaz de detectar cinco estados de ánimo diferentes (Muy desanimado, Desanimado, Normal, Animado, Muy animado) desde el robot y en tiempo real.

El quinto objetivo, que era desarrollar un sistema de monitorización de la vida diaria de una persona, se logró con el desarrollo de la arquitectura software de SARDAM. Se integraron todos los sistemas que se obtuvieron en los objetivos previos y funcionalidades útiles de los módulos de NAOqi para poder implementar las tareas y actividades del SAR. Además, SARDAM se diseñó de manera que permite añadir, eliminar y modificar tareas y actividades sin ninguna complejidad; incluso facilita usar otro robot diferente a NAO al desacoplar los

métodos de NAOqi del sistema. Es un sistema mantenible, modular y reutilizable.

El último de los objetivos era realizar una serie de pruebas con el robot NAO y usuarios reales. Para ello se crearon los escenarios que cubrían diferentes personalidades que se podían dar en la interacción humano-robot en el caso de este proyecto. En dichas pruebas se consiguieron resultados lo suficientemente satisfactorios, obteniendo una reacción positiva por parte de los usuarios en cada escenario.

En este proyecto se aprendió, en primer lugar, cómo utilizar un robot humanoide y conocer su funcionamiento. También se hizo una introducción a la robótica de asistencia social, para qué sirve y como son y funcionan los robots de este campo. Se aprendió a diseñar y desarrollar desde cero un sistema de este tipo, obteniendo un prototipo inicial que contiene las bases de lo que a un sistema SAR se refiere. Además, se conocieron las diferentes tecnologías de visión artificial de reconocimiento y detección, integrándolas en el proyecto y utilizándolas para, incluso, obtener otro sistema de detección.

En conclusión, se puede considerar que SARDAM es el punto de arranque de una línea con gran potencial basada en el uso de robots como elemento clave de interacción.

7.1 Líneas futuras

SARDAM actualmente es un prototipo básico de lo que puede llegar a ser un sistema de monitorización. Este trabajo abre un abanico de posibilidades de mejora y ampliación de las funcionalidades del sistema desarrollado. En el futuro se podrían integrar cuestiones como las que se exponen a continuación:

- Mejorar el sistema de detección de objetos, de manera que pueda detectar objetos más relacionados con este proyecto. Por ejemplo, objetos como llaves, pastillas, carteras, gafas, etc.
- Mejorar las tareas y actividades implementadas. Algunas de estas mejoras pueden ser:
 - Implementar más ejercicios de movilidad física para tener más variedad de rutinas. Además, que se pueda controlar si la persona está haciendo dichos ejercicios correctamente.
 - Desarrollar más juegos cognitivos para que dicha tarea sea mucho más variada.
 - Mejorar la actividad de reminiscencia actual, de manera que el robot pueda recordarle al usuario todos los eventos pasados de su vida y detecte los objetos que le enseña. Que el robot tenga la capacidad de mantener una conversación en base a anteriores con la persona.
 - A la hora de hacer de comer, que el robot tenga la capacidad de recitarle las recetas en caso de que la persona no se acuerde como se hacían.

- A la hora de tomar pastillas, que el robot pueda decirle cuáles son las pastillas que debe tomarse.
- Integrar sensores en casa para mejorar la monitorización de las tareas. Por ejemplo, se podría tener un sensor en la puerta del baño que controle cuando la persona entra o sale de él.
- Integrar un sistema de detección de caídas.
- Mejorar la detección del estado emocional, en el que se observen más características que las obtenidas por imagen y audio.
- Implementar replanificación del horario. Es decir, que se pueda cambiar la hora de la tarea en caso de, por ejemplo, el usuario no quiera hacerla en la hora establecida ese día. También se podría adaptar el horario a las preferencias del usuario, de manera que la rutina sea mucho más dinámica y aumentando las posibilidades de que el usuario esté más animado para hacer las tareas.
- Permitir que se pueda monitorizar a más de una persona.
- Utilizar algoritmos de Aprendizaje automático que, en función de los datos obtenidos de la interacción con el usuario, permitan adoptar las rutinas diarias al modo de vida, preferencias y peculiaridades de este, tal como haría un experto humano.

Apéndices

Documentación sobre las AVD

ESTE apéndice contiene los tres documentos que la psicóloga aportó sobre las actividades de la vida diaria de una persona mayor (AVD) y el protocolo de actuación que sigue el centro de mayores SUACASA, aunque este último no es de mucha utilidad para este TFG ya que va dirigido para las personas mayores que ya no podrían valerse por sí mismas. El más importante de todos es el último documento, llamado "Actividades de la vida diaria (AVD)", que trata sobre las AVD, su clasificación en tres niveles y recomendaciones para su correcta realización.

INDICE DE BARTHEL. Actividades básicas de la vida diaria

ESCALA + HABILIDADES
PARA
DETERMINAR
AUTONOMÍA
DE CADA
PERSONA

Parámetro	Situación del paciente	Puntuación
Total:		
Comer	- Totalmente independiente	10
	- Necesita ayuda para cortar carne, el pan, etc.	5
	- Dependiente	0
Lavarse	- Independiente: entra y sale solo del baño	5
	- Dependiente	0
Vestirse	- Independiente: capaz de ponerse y de quitarse la ropa, abotonarse, atarse los zapatos	10
	- Necesita ayuda	5
	- Dependiente	0
Arreglarse	- Independiente para lavarse la cara, las manos, peinarse, afeitarse, maquillarse, etc.	5
	- Dependiente	0
Deposiciones (valórese la semana previa)	- Continencia normal	10
	- Ocasionalmente algún episodio de incontinencia, o necesita ayuda para administrarse supositorios o lavativas	5
	- Incontinencia	0
Micción (valórese la semana previa)	- Continencia normal, o es capaz de cuidarse de la sonda si tiene una puesta	10
	- Un episodio diario como máximo de incontinencia, o necesita ayuda para cuidar de la sonda	5
	- Incontinencia	0
Usar el retrete		
	- Independiente para ir al cuarto de aseo, quitarse y ponerse la ropa...	10

INDICE DE BARTHEL. Actividades básicas de la vida diaria

Parámetro	Situación del paciente	Puntuación
	- Necesita ayuda para ir al retrete, pero se limpia solo	5
	- Dependiente	0
Trasladarse	- Independiente para ir del sillón a la cama	15
	- Mínima ayuda física o supervisión para hacerlo	10
	- Necesita gran ayuda, pero es capaz de mantenerse sentado solo	5
	- Dependiente	0
Deambular	- Independiente, camina solo 50 metros	15
	- Necesita ayuda física o supervisión para caminar 50 metros	10
	- Independiente en silla de ruedas sin ayuda	5
	- Dependiente	0
Escalones	- Independiente para bajar y subir escaleras	10
	- Necesita ayuda física o supervisión para hacerlo	5
	- Dependiente	0

Máxima puntuación: 100 puntos (90 si va en silla de ruedas)

Resultado	Grado de dependencia
< 20	Total
20-35	Grave
40-55	Moderado
≥ 60	Leve
100	Independiente

PROTOCOLO DE ORIENTACIÓN

Me gustaría agradecer por adelantado la colaboración, que estoy convencida, manifestará todo el personal del centro, apelando a la profesionalidad que demuestran día tras día. Estas indicaciones son una primera guía dentro del programa de atención psicológica que se irá implantando en el centro. En principio se refiere a la relación con los internos con un mayor deterioro cognitivo, en un intento de mantener las habilidades y capacidades que conservan. Esto, como es lógico es nuestro primordial y último objetivo.

Señalaré unas pequeñas indicaciones para establecer un criterio común, (o sea lo que se pretende es que todo el personal haga lo mismo).

- ❖ Aprovechar las actividades cotidianas para realizar este tipo de preguntas, sobre todo la hora del baño, aseo (adaptar las preguntas al tiempo que se tenga, en principio se harán menos y poco a poco con un entrenamiento se logrará a hacer más)
- ❖ Escoger palabras sencillas y frases cortas, utilizando un tono de voz amable y tranquila.
- ❖ Tratar de presentar las preguntas e instrucciones de una manera positiva.
- ❖ Prestar atención a las preocupaciones de la persona, aunque sea difícil entenderla. Ser amable y respetuoso. Tener paciencia y calma.
- ❖ Decir a la persona lo que usted va a hacerle, paso por paso, y permítale hacer por sí misma todo lo que le sea posible.

PROTOCOLO DE PREGUNTAS PARA LOS INTERNOS

- ❖ ORIENTACIÓN TEMPORAL.
 - ¿Qué día de la semana es hoy?
 - ¿Qué mes estamos?
 - ¿En qué momento del día estamos si vamos a desayunar, comer, cenar? (según la hora que sea)
- ❖ ORIENTACIÓN ESPACIAL
 - ¿Sabe dónde estamos? Estamos en la residencia
 - ¿Con quién comparte su habitación?
 - ¿En qué ciudad estamos?
 - ¿En qué país vivimos?
- ❖ ORIENTACIÓN DE LA PERSONA
 - ¿Cómo se llama?
 - ¿Cómo se llama su marido?
 - ¿Cómo se llaman sus hijos, sobrino, hermanos, primos? (elegir las personas más allegadas)

❖ EJERCITACIÓN DE CAPACIDADES LINGÜÍSTICAS.

- aprovechar las tareas cotidianas para ejercitar sus capacidades lingüísticas, por ejemplo; cuando se bañe que denomine los objetos que hay en el baño, etc.
- realizar tareas de lectura (lectura diaria de prensa, libros, etc...) se deben cerciorar de que todo el mundo escuche y a ser posible sentarse cerca de Esther. Se debería intentar implicar en diálogo en relación a la noticia a las personas más reacias poco a poco (aprovechando noticias que tengan alguna relación con lo vivido por ellos)

❖ MEMORIA REMOTA.

- ¿Qué recuerda de su trabajo? ¿En qué consistía?
- ¿Se acuerda de algún refrán popular? Si no se acuerda por ejemplo decirle la auxiliar uno. Ej.: A quien madruga dios le ayuda. Y a lo largo del día, por ejemplo en las comidas preguntarle por el refrán.

Actividades de la vida diaria (AVD)

El ser humano es un ser activo por naturaleza que ocupa su tiempo en actividades encaminadas a responder sus necesidades y deseos.

Las personas se realizan a través de sus ocupaciones en interacción constante con el ambiente, de ahí la importancia de la realización de actividades que sean significativas y gratificantes para ellas.

La salud y la ocupación están vinculadas porque la pérdida de salud disminuye la capacidad para comprometerse en la actividad.

En este caso deberemos de adaptar la actividad teniendo en cuenta las características individuales de cada caso, sus motivaciones, gustos e intereses.

Las actividades de la vida diaria (AVD) son un conjunto de tareas o conductas que una persona realiza de forma diaria y que le permiten vivir de forma autónoma e integrada en su entorno y cumplir su rol o roles dentro de la sociedad.

Varían entre las actividades que son más fundamentales para la supervivencia como la alimentación, hasta algunas tareas más complejas del autocuidado personal e independencia vital, como hacer la compra, cocinar o usar el teléfono. Así mismo todas las actividades relacionadas con el ocio, el tiempo libre y el esparcimiento, y aquellas relacionadas con el ámbito laboral.

Todas las personas realizan las AVD en un grado, forma e importancia que varía de un individuo a otro, con una gran influencia de su cultura.

Planificar actividades de la vida diaria

Antes de planificar las actividades que vamos a realizar con la persona dependiente intentaremos definir y conocer las características de éstas:

Tipo de actividades:

- Actividades básicas de la vida diaria (ABVD)
- Actividades instrumentales de la vida diaria (AIVD)
- Actividades avanzadas de la vida diaria (AAVD)

Actividades básicas de la vida diaria (ABVD)

- Son universales
- Están ligadas a la supervivencia y condición humana, a las necesidades básicas de cada individuo.
- Están dirigidas a uno mismo
- Se realizan cotidianamente y de forma automática

Dentro de las ABVD se incluyen:

- Alimentación
- Aseo
- Baño
- Control de esfínteres
- Vestido
- Movilidad personal
- Sueño y descanso

Actividades instrumentales de la vida diaria (AIVD)

- Conllevan un mayor sesgo cultural
- Están ligadas al entorno
- Son un medio para obtener o realizar otra acción
- Suponen una mayor complejidad cognitiva y motriz e implican la interacción con el medio más inmediato.

Dentro de las AIVD se incluyen:

- Utilizar distintos sistemas de comunicación (escribir, hablar por teléfono...)
- Movilidad comunitaria (conducir, uso de medios de transporte)
- Mantenimiento de la propia salud
- Manejo de dinero (compras)

No tiene por qué ^{tu}proyecto enfocarse solo a mayores sino tb. a discapacitados, personas que han sufrido ictus, etc.

SACADO DE
cocomfecyl.es
COCEMFE
(Confederación de
Personas con Discapa-
cidad Física y
Orgánica)

- Establecimiento y cuidado del hogar
- Cuidado de otros
- Uso de procedimientos de seguridad
- Respuesta ante emergencias.

Actividades avanzadas de la vida diaria (AAVD)

- No son indispensables para el mantenimiento de la independencia
- Están en relación con el estilo de vida del sujeto.
- Permiten al individuo desarrollar sus papeles dentro de la sociedad

Dentro de las AAVD se incluyen:

- Educación
- Trabajo
- Ocio
- Participación en grupos
- Contactos sociales
- Viajes
- Deportes

¿Por qué se afecta la realización de estas actividades?

Cada ocupación pone en juego una serie de habilidades o destrezas humanas fundamentales, que son necesarias para la realización de las diferentes actividades la vida diaria. La imposibilidad de llevarlas a cabo por uno mismo, puede estar condicionado por un déficit cognitivo-conductual, tanto como por déficits motores.

La dificultad para llevar a cabo las actividades de la vida diaria está directamente relacionada con el grado de afectación de los componentes de ejecución. Veamos a continuación algunos de los déficits que podemos observar en personas en situación de dependencia:

A nivel cognitivo:

Alteraciones en memoria, procesos atencionales, orientación de tiempo, espacio y persona, capacidad de aprendizaje, secuenciación de acciones, planificación y organización de tareas, capacidad de iniciar y finalizar una tarea...

A nivel psicopatológico:

Labilidad emocional, apatía, impulsividad, rigidez mental...

A nivel motor podemos observar:

Aumento o disminución del tono muscular, pérdida de amplitud de movimientos, disminución de la fuerza, temblor, alteraciones de la sensibilidad...

Recomendaciones para mantener la autonomía de las personas en situación de dependencia

La persona debe ser estimulada para realizar las AVD de la forma más independiente posible. Por ello alentar a la persona en situación de dependencia a realizar la actividad por sí mismo debe ser prioritario.

Es fundamental apoyar el nivel de ocupación de la persona teniendo en cuenta los problemas que aparecen en cada enfermedad y en cada fase de esta.

Mantener el funcionamiento ocupacional es la mejor arma contra la inactividad, sedentarismo, sentimientos de inutilidad, alteración de conductas, desesperanza... Continuar realizando actividades en la medida que sea posible, contribuye a la orientación, establecimiento de rutinas, disminución de conductas disruptivas, fomentar el sentimiento de utilidad y conservar la identidad personal.

Con todo ello contribuiremos a mantener la independencia y en definitiva aumentar la calidad de vida, tanto de la persona en situación de dependencia como de sus cuidadores.

Veremos una serie de estrategias que nos pueden ayudar a mantener la autonomía, siempre dependiendo de las diferentes afectaciones a nivel cognitivo, psicopatológico y/o motor que podamos encontrar:

- **Simplificar y secuenciar las tareas:** Facilitan la realización de la actividad y proporcionan mayor confort y comodidad al afectado y a los cuidadores. Permiten conservar la energía y establecen prioridades en el día a día, con periodos de descanso y evitando tareas repetitivas.
- **Mantenimiento de un posicionamiento adecuado:** Mejora la funcionalidad de los brazos y compensa la falta de fuerza de movimientos en contra de la gravedad.
- **Dispositivos de apoyo:** Conjunto de recursos de ayuda que facilitan la realización de la actividad, disminuyendo la carga y los requerimientos físicos para realizarla.
- **Modificación del entorno:** adaptando los diferentes espacios en función de las necesidades y el grado de deterioro.
- **Colaboración de la persona:** Si la persona en situación de dependencia no es capaz de realizar la tarea de forma autónoma, siempre que sea posible, debemos facilitar el acceso a las tareas que sí es capaz de hacer, aunque tarde más tiempo del habitual. Para ello, podemos ofrecerle apoyo verbal o pequeñas instrucciones.

Ocio y tiempo libre

Tanto las personas con discapacidad como sus cuidadores deben conocer que existen recursos que pueden seguir dando continuidad a su recuperación y que fomenten el ocio y las actividades de esparcimiento.

Es importante entender y aprender a disfrutar del tiempo libre como medio para sentirse mejor, prevenir enfermedades, aumentar la creatividad y proporcionar una mejor calidad de vida.

Para ello debemos aprender a organizar el tiempo libre de forma adecuada para que resulte provechoso.

Conlleva mayor beneficio si las actividades de ocio se comparten con otras personas, ya sean familiares, amigos, compañeros, etc...

Algunas de las actividades que se pueden tener en cuenta:

- Realización de ejercicio físico o deporte, adaptado y en consonancia a las características de cada persona. Aporta una mejora de la salud y de la calidad de vida. Ayuda a retrasar la aparición de enfermedades y evitar el sobrepeso.
- Realización de alguna actividad mental, lectura de libros, crucigramas, sopas de letras...
- Acudir a talleres que organicen en la comunidad: taller de pintura, de teatro, de manualidades, música, fotografía...
- Conocer otros lugares a través de excursiones y vacaciones

Recomendaciones generales para los cuidadores

- - Mantener un horario y rutina constantes que permita automatizar las tareas.
- - Realizar una secuencia de pasos simples y constantes para la realización de AVD.
- - Mantener la regularidad del ciclo sueño-vigilia.
- - Establecer un entorno seguro, estable, sencillo, facilitador, orientador y adaptado.
- - No sobreproteger en exceso, estimular al máximo su autonomía.
- - Adaptar cada tarea gradualmente en función del deterioro de la enfermedad.
- - No proporcionar ayudas o apoyos que no necesiten por el tipo o fase de la enfermedad.
- - Estimular y mantener capacidades conservadas del individuo.
- - Estimulación de aspectos cognitivos de la persona.
- - Simplificar al máximo las tareas para permitir la participación de la persona, dando si es necesario, instrucciones de forma oral para cada uno de los pasos.
- - Estimular el habla y cualquier comunicación con el exterior. Emitir enunciados sencillos, claros y audibles.
- - Mantener orientación temporal, espacial y de persona.
- - Respetar intimidad y preferencias

- - Dar tiempos de tranquilidad y relajación tanto para el cuidador como para la persona con discapacidad.
- - Realizar actividad física adaptada tan frecuente como sea posible.
- - Buscar y realizar actividades de ocio y tiempo libre más motivadoras y participativas para el usuario.
- - Estar en contacto con asociaciones o centros para personas con discapacidad, que puedan proporcionar asesoramiento en las diferentes etapas.
- - Ser flexibles y pacientes.

Descripción de las tareas

EN este apéndice se continúa con la descripción de las tareas que se inició en la Sección 5.6.2.

Tarea **ComidaMediodia**:

- Prioridad: 1
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para comer.
 2. Se registra en el fichero de monitorización la hora en ese momento.
 3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
 4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
 5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **Merienda**:

- Prioridad: 1
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para merendar.
2. Se registra en el fichero de monitorización la hora en ese momento.
3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **Cena:**

- Prioridad: 1

- Sensores: HumanTracked, Emotion.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para cenar.
2. Se registra en el fichero de monitorización la hora en ese momento.
3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **Acostarse:**

- Prioridad: 1

- Sensores: HumanTracked, Emotion.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot contesta diciendo “Muy bien” y dice hasta que hora tiene el usuario para comer.
2. Se registra en el fichero de monitorización la hora en ese momento.
3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **LimpiezaBasica:**

- Prioridad: 2

- Sensores: HumanTracked, Emotion.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot le sugiere que pase la aspiradora y, si es necesario, que ponga la lavadora. También le avisa a que hora termina la tarea.
2. Espera hasta que termine el tiempo de la tarea.
3. La tarea finaliza.

Tarea **SalirCompra:**

- Prioridad: 2

- Sensores: HumanTracked, Emotion.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot contesta diciendo “Muy bien”, le recuerda que lleve la cartera y las llaves, y le dice hasta que hora tiene el usuario para hacer la compra.
2. Se registra en el fichero de monitorización la hora en ese momento.

-
3. Se esperan 20 segundos para dar tiempo al usuario para que desaparezca de la vista del robot (y vaya a desayunar), y a partir de ahí se sigue esperando pero controlando con *hay_personas()* si vuelve a ver al usuario. Esto es para que sirva de información (de manera general) de cuánto tiempo el usuario ha podido estar desayunando.
 4. Si ve al usuario, lo registra en el fichero log de monitorización (monitorizacion.log).
 5. La tarea termina cuando el tiempo designado para ella finaliza.

Tarea **HacerComer**:

- Prioridad: 2
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta diciéndole “Muy bien” y a que hora debería de haber terminado.
 2. Espera hasta que termine el tiempo de la tarea.
 3. La tarea finaliza.

Tarea **LimpiezaCocina**:

- Prioridad: 2
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot le recuerda que se encargue de la lavadora también. Le avisa a que hora termina la tarea.
 2. Espera hasta que termine el tiempo de la tarea.
 3. La tarea finaliza.

Tarea **JuegoCognitivo**:

- Prioridad: 2
- Sensores: HumanTracked, Emotion, Google SpeechRecognition.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. Se calcula la duración que tendrá la actividad. Esto se hace calculando el tiempo que queda desde ese momento a la hora final, que será lo que dure la actividad.
2. El robot contesta diciendo “Bien. Te propongo este juego cognitivo”.
3. Se ejecuta la actividad JuegoCognitivo con la duración estimada.
4. Una vez finalizada la actividad, la tarea termina.

Tarea **Reminiscencia:**

- Prioridad: 2

- Sensores: HumanTracked, Emotion, Yolo.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder, ALVideoDevice.

- Acciones:

1. Se calcula la duración que tendrá la actividad. Esto se hace calculando el tiempo que queda desde ese momento a la hora final, que será lo que dure la actividad.
2. El robot contesta diciendo “Muy bien. Vamos a hacer esta actividad de reminiscencia”.
3. Se ejecuta la actividad LibroDeVida con la duración estimada.
4. Una vez finalizada la actividad, la tarea termina.

Tarea **Llamada:**

- Prioridad: 2

- Sensores: HumanTracked, Emotion.

- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.

- Acciones:

1. El robot contesta diciéndole “Muy bien” y le dice hasta que hora tiene para realizar la llamada.
2. Espera hasta que termine el tiempo de la tarea.
3. La tarea finaliza.

Tarea **TomarMedicacion:**

- Prioridad: 2

-
- Sensores: HumanTracked, Emotion.
 - Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
 - Acciones:
 1. El robot contesta diciéndole “Muy bien” y que ya sabe dónde se encuentran los medicamentos.
 2. Espera hasta que termine el tiempo de la tarea.
 3. La tarea finaliza.

Tarea **DescansoOcio**:

- Prioridad: 3
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta “Muy bien” al usuario y le dice a que hora termina la actividad.
 2. Espera hasta que termine el tiempo de la tarea.
 3. La tarea finaliza.

Tarea **SalirExterior**:

- Prioridad: 3
- Sensores: HumanTracked, Emotion.
- Actuadores: ALTextToSpeech, ALAudioRecorder, ALVideoRecorder.
- Acciones:
 1. El robot contesta “Bien” al usuario y le dice a que hora termina la actividad.
 2. Espera hasta que termine el tiempo de la tarea.
 3. La tarea finaliza.

Lista de acrónimos

SAR *Socially Assistive Robotics.*

UDC *Universidad de A Coruña.*

IA *Inteligencia Artificial.*

API *Application Programming Interface.*

RFID *Radio Frequency Identification.*

SVM *Support Vector Machine.*

AVD *Actividades de la Vida Diaria.*

ABVD *Actividades Básicas de la Vida Diaria.*

AIVD *Actividades Instrumentales de la Vida Diaria.*

AAVD *Actividades Avanzadas de la Vida Diaria.*

Glosario

Espacio cartesiano Representación del espacio donde la posición de un punto queda unívocamente definida por su distancia a n planos ortogonales entre sí, quedando esa posición expresada por n números reales llamados coordenadas, y siendo el origen de coordenadas el punto en el que cortan los n planos.

Reminiscencia Acción de representarse u ofrecerse a la memoria el recuerdo de algo que pasó.

Actividades de la vida diaria Conjunto de tareas o conductas que una persona realiza de forma diaria y que le permiten vivir de forma autónoma e integrada en su entorno y cumplir con su rol o roles dentro de la sociedad.

Gerontogimnasia Gimnasia enfocada a la tercera edad con el objetivo de mantener el estado de salud y prevenir el decline funcional de las personas mayores. Se trata de ejercicios destinados a trabajar el movimiento de músculos y articulaciones con el fin último de mejorar el aparato cardiorrespiratorio y las funciones cognitivas.

Bibliografía

- [1] D. Feil-Seifer and M. Matarić, “Defining socially assistive robotics,” *Proceedings of the IEEE 9th International Conference on Rehabilitation Robotics*, vol. 2005, pp. 465 – 468, 07 2005.
- [2] J. Broekens, M. Heerink, and H. Rosendal, “Assistive social robots in elderly care: A review,” *Gerontechnology*, vol. 8, pp. 94–103, 04 2009.
- [3] J. Abdi, A. Al-Hindawi, T. Y. Ng, and M. P. Vizcaychipi, “Scoping review on the use of socially assistive robot technology in elderly care,” *BMJ Open*, vol. 8, 2018.
- [4] A. K. Ostrowski, D. DiPaola, E. Partridge, H. W. Park, and C. Breazeal, “Older adults living with social robots: Promoting social connectedness in long-term communities,” *IEEE Robotics Automation Magazine*, vol. 26, no. 2, pp. 59–70, 2019.
- [5] Jibo, “Jibo robot.” [En línea]. Disponible en: <https://www.jibo.com/>
- [6] S. Robotics, “Nao the humanoid and programmable robot.” [En línea]. Disponible en: <https://www.softbankrobotics.com/emea/en/nao>
- [7] PARO, “Paro therapeutic robot.” [En línea]. Disponible en: <http://www.parorobots.com/>
- [8] Aibo, “aibo.” [En línea]. Disponible en: <https://us.aibo.com/>
- [9] M. Pollack, L. Brown, D. Colbry, C. Orosz, B. Peintner, S. Ramakrishnan, S. Engberg, J. Matthews, J. Dunbar-Jacob, C. McCarthy, M. Montemerlo, and J. Pineau, “Pearl: A mobile robotic assistant for the elderly,” 06 2002.
- [10] S. Coşar, M. Fernández-Carmona, R. Agrigoroaie, J. Pages, F. Ferland, F. Zhao, S. Yue, N. Bellotto, and A. Tapus, “Enrichme: Perception and interaction of an assistive robot for the elderly at home,” *International Journal of Social Robotics*, 02 2020.
- [11] S. Coradeschi, A. Cesta, G. Cortellessa, L. Coraci, J. González-Jiménez, L. Karlsson, F. Furfari, A. Loutfi, A. Orlandini, and F. Palumbo, “Giraffplus: combining social interaction

- and long term monitoring for promoting independent living,” *2013 6th International Conference on Human System Interactions, HSI 2013*, pp. 578–585, 06 2013.
- [12] S. Coradeschi, A. Cesta, G. Cortellessa, L. Coraci, C. Galindo, J. González-Jiménez, L. Karlsson, A. Forsberg, S. Frennert, F. Furfari, A. Loutfi, A. Orlandini, F. Palumbo, F. Pecora, S. Von Rump, A. Štimec, J. Ullberg, and B. Ötslund, “Giraffplus: A system for monitoring activities and physiological parameters and promoting social interaction for elderly,” *Advances in Intelligent Systems and Computing*, vol. 300, pp. 261–271, 07 2014.
- [13] S. Economy, “European union: Giraffplus project combines robotics and home automation to take care of the elderly.” [En línea]. Disponible en: <http://www.silvereco.org/en/european-union-giraffplus-project-combines-robotics-and-home-automation-to-take-care-of-the-elderly/>
- [14] BALIDEA, “Neta v3a.” [En línea]. Disponible en: <https://balidea.com/es/portfolio/neta-v3a>
- [15] A. Robotics, “Stevie the robot.” [En línea]. Disponible en: <https://stevietherobot.com/>
- [16] T. C. Dublin, “Design of embodied social robot v2.0 ‘stevie’.” [En línea]. Disponible en: <https://www.tcd.ie/mecheng/research/robotics/projects/stevie.php>
- [17] Zorabots, “Zorabots.” [En línea]. Disponible en: <https://www.zorarobotics.be/>
- [18] Nao, “Zorabots.” [En línea]. Disponible en: <https://zorarobotics.be/robots/nao>
- [19] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” pp. 815–823, 2015.
- [20] M. L. Mastery, “How to develop a face recognition system using facenet in keras.” [En línea]. Disponible en: <https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/>
- [21] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018. [En línea]. Disponible en: <http://arxiv.org/abs/1804.02767>
- [22] M. L. Mastery, “How to perform object detection with yolov3 in keras.” [En línea]. Disponible en: <https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/>
- [23] A. Kumar, “Faceemotion_id.” [En línea]. Disponible en: https://github.com/abhijeet3922/FaceEmotion_ID

- [24] O. Arriaga, M. Valdenegro-Toro, and P. Plöger, “Real-time convolutional neural networks for emotion and gender classification,” *CoRR*, vol. abs/1710.07557, 2017.
- [25] PyPI, “Speechrecognition.” [En línea]. Disponible en: <https://pypi.org/project/SpeechRecognition/>
- [26] A. Robots, “Consulta nuestros precios.” [En línea]. Disponible en: <https://aliverobots.com/precio/>
- [27] Indeed, “Salarios para empleos de desarrollador/a junior en españa.” [En línea]. Disponible en: <https://es.indeed.com/salaries/desarrollador-junior-Salaries>
- [28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, vol. 23, 04 2016.
- [29] Vokaturi, “Vokaturi emotion recognition software.” [En línea]. Disponible en: <https://vokaturi.com/>
- [30] Rockikz, “Speech emotion recognition.” [En línea]. Disponible en: <https://github.com/x4nth055/emotion-recognition-using-speech>
- [31] M. Puthran, “Speech emotion analyzer.” [En línea]. Disponible en: <https://github.com/MITESHPUTHRANNEU/Speech-Emotion-Analyzer>
- [32] M. P. J. S. M. D. M. D. F. J. D. S. Enrique G. Fernández-Abascal, Beatriz García Rodríguez, *Psicología de la emoción*, 1st ed. Editorial universitaria Ramón Areces, 2010.
- [33] W. Hashim Abdulsalam, d. r. al hamdani, and M. Al Salam, “Emotion recognition system based on hybrid techniques,” *International Journal of Machine Learning and Computing*, vol. 9, pp. 490–495, 07 2019.
- [34] S. R. Livingstone and F. A. Russo, “The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS),” Apr. 2018, Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak. [En línea]. Disponible en: <https://doi.org/10.5281/zenodo.1188976>
- [35] F. M. Archive, “Free music archive.” [En línea]. Disponible en: <https://freemusicarchive.org/>

